MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.CC | July 2023 | Issue #68

DIY SMART CAMERA

PROGRAM YOUR OWN PHOTOGRAPHY MACHINE



Recycle PLA

Turn 3D printer waste into new projects

Tim Hunkin

The secret art of artisanal arcade machines

Planetary gears

Design your own gearbox

Off-grid power tools

Keep building when there's no sockets

July. 2023
Issue #68 £6

9 772515 514006 68>

HEXAGONS **LEDS** MACRAMÉ **E-INK**

Free eBook!



Download your copy from

hsmag.cc/freecadbook



Welcome to HackSpace magazine

Most people receive the majority of their information about the world through their eyes. This is great, but they do have a few flaws. You can't adjust the frame rate on them, they need to be switched off for around eight hours a day, and there's no record function on them.

Fortunately, inventive scientists have come up with a solution for this – cameras. These let you see the world in ways impossible with our on-board faculties. While there are some excellent off-the-shelf cameras available, you can get much more control by building your own, and there's one standout platform for this: Raspberry Pi.

In this issue, we're looking at how you can combine programming with a range of camera modules to capture the world in ways our primitive face-holes simply can't manage. The results are fun, beautiful, and give us a new perspective on the world (literally).

BEN EVERARD
Editor ben.everard@raspberrypi.com

Got a comment, question, or thought about HackSpace magazine?

get in touch at **hsmag.cc/hello**

GET IN TOUCH

hackspace@raspberrypi.com

hackspacemag

hackspacemag

ONLINE

hsmag.cc

Available on the App Store

GET IT ON Google Play



PAGE 30
FREE PICO W
WHEN YOU SUBSCRIBE

EDITORIAL
Editor
Ben Everard
ben.everard@raspberrypi.com

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.com

Sub-Editors
David Higgs, Nicola King

DESIGN
Critical Media and Raspberry Pi
criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Sam Ribbits, Olivia Mitchell, Sara Parodi, Jack Willis

Photography
Brian O'Halloran

CONTRIBUTORS
Marc de Vinck, Andrew Lewis, Rob Miles, Jo Hinchliffe, Nicola King, Phil King

PUBLISHING
Publishing Director
Brian Jepson
brian.jepson@raspberrypi.com

Advertising
Charlie Milligan
charlotte.milligan@raspberrypi.com

DISTRIBUTION
Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS
Unit 6, The Enterprise Centre,
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE

To subscribe
01293 312189
hsmag.cc/subscribe

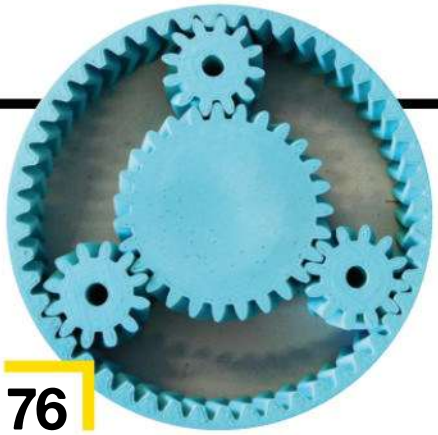
Subscription queries
hackspace@subscriptionhelpline.co.uk

recycle

This magazine is printed on paper sourced from sustainable forests. The printer operates an environmental management system which has been assessed as conforming to ISO 14001.

HackSpace magazine is published by Raspberry Pi Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2515-5148.

Contents



76

06 SPARK

06 **Top Projects**
Oh my, you makers are clever

16 **Objet 3d'art**
Like Bitcoin, but fewer bits and more metal

18 **Letters**
Don't fear the Furby

21 LENS

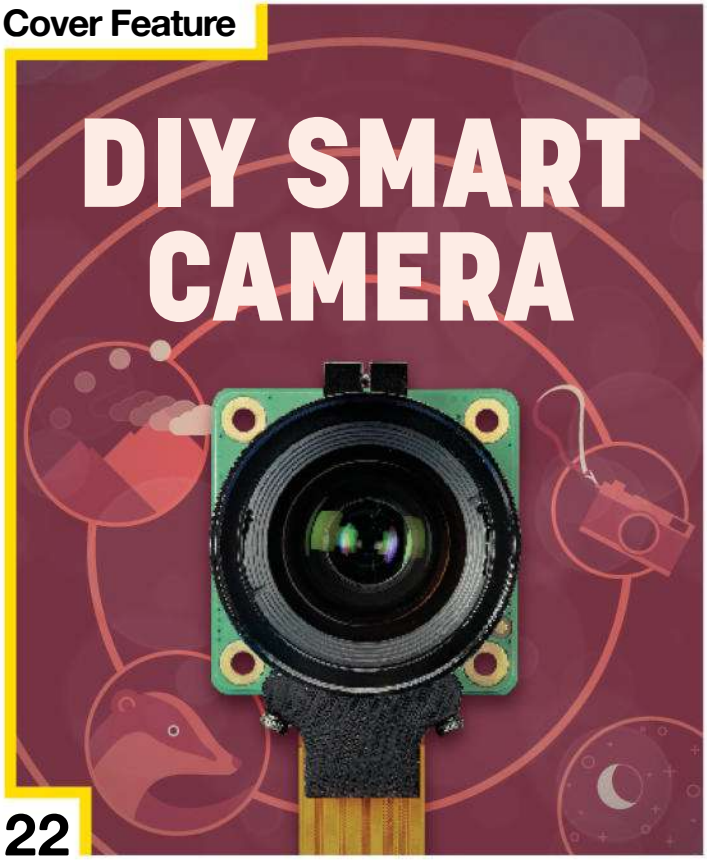
22 **Photography**
Now you can buy a Raspberry Pi, here's how to use it

32 **How I Made:** Off-grid power tools
Recharge batteries your way

38 **Interview:** Tim Hunkin
Inside a secret life

46 **In the workshop**
Reverse-engineering an LED display

Cover Feature



22

Tutorial

Bluetooth keyboard



70 Customise your wireless typing rig



08



32



86

Tutorial

Recycling



58 Save our precious planet one failed print at a time

Interview

Tim Hunkin



38 Tim helped a generation of makers peek inside hidden worlds

51 FORGE

52 **SoM** KiCad
Taking your first steps with RP2040

58 **Tutorial** Recycling PLA
Squeeze and squish your old prints into new things

64 **Tutorial** Beginning microcontrollers
Make electrons follow instructions

68 **Tutorial** Polycarbonate
The most futuristic-sounding printer filament

70 **Tutorial** Bluetooth keyboard
Because you're too cool for off-the-shelf typing

76 **Tutorial** Planetary gears
Build a drive chain that's out of this world

82 **Tutorial** Macramé
Like 3D printing, but it's meant to look like spaghetti

85 FIELD TEST

86 **Best of Breed**
The newest new things that are newly released

92 **Review** Hextraction
Waste even more time with your 3D printer

96 **Crowdfunding** Wyldcard
Code your own e-paper card games

Some of the tools and techniques shown in HackSpace Magazine are dangerous unless used with skill, experience and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. HackSpace Magazine is intended for an adult audience and some projects may be dangerous for children. Raspberry Pi Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in HackSpace Magazine. Laws and regulations covering many of the topics in HackSpace Magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in HackSpace Magazine may go beyond. It is your responsibility to understand the manufacturer's limits. HackSpace magazine is published monthly by Raspberry Pi Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS, United Kingdom. Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701, is the mailing agent for copies distributed in the US and Canada. Application to mail at Periodicals prices is pending at Williamsport, PA. Postmaster please send address changes to HackSpace magazine c/o Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701.

REGULAR

Mechanical Apple Watch

By Jack Spiggle

hsmag.cc/NanoRobotGeek

Mechanical watches aren't just timepieces: they're miniature works of art, engineering marvels. They don't run out of battery power and, if you look after them, they can last for decades. This author has a cheap pocket-watch made to the famously high manufacturing standards of the late USSR, and it's still functioning to this day.

The natural thing to do with an old, useless Apple Watch then, is to replace its obsolete electronics, unsupported software, and drained battery with a mechanism that'll tell the time without constantly annoying you with updates from social media. That's what Jack Spiggle has done with this build, augmenting a 42 mm Apple Watch Series 1 in stainless steel with a Seiko NH38 watch movement. Jack's donor watch had been unsupported by Apple since 2020, and the battery life was down to just four hours – he's rescued it from the scrap-heap and put it to brilliant use.



Right
We prefer the dumb watch to the smart watch


SWATCH • 42MM CASE • 316L STAINLESS STEEL • SAPPHIRE CRYSTAL • CERAMIC BACK • SERIAL FHEPHAD100698
TWENTY-FOUR JEWELS
NH38

REGULAR

Four-shaft loom

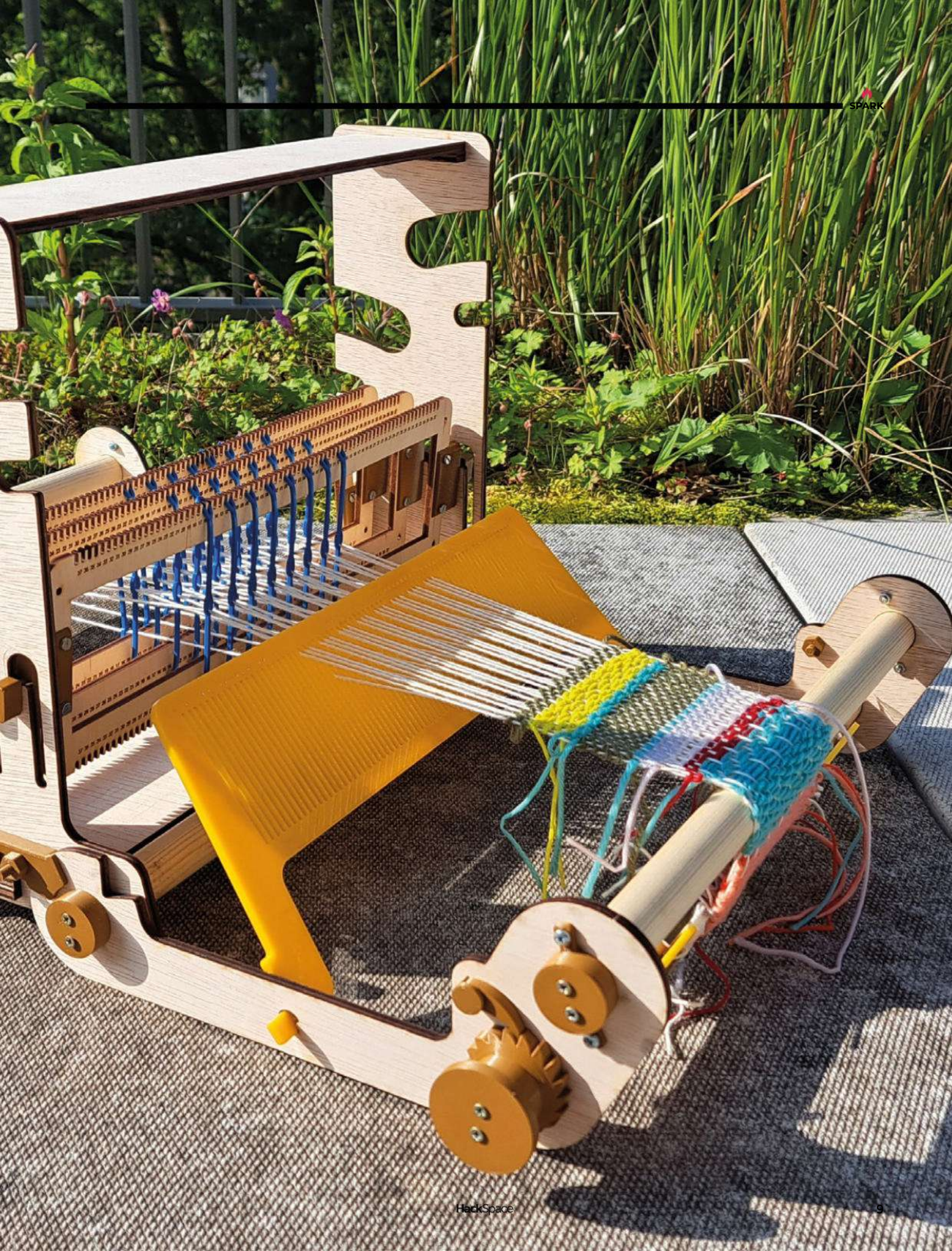
By **Asli Aydin Aksan** hsmag.cc/4ShaftLoom

Weaving is a complicated business; so much so that you could plausibly argue that computer programming was invented as a by-product of encoding weaving designs. This design, by architect, researcher, and maker Asli Aydin Aksan, is a more complicated build than some other homemade looms we've seen, but the brilliance of that is that it enables the user to create more complex patterns than you would if you were using a simpler loom.

What we love about this is that it's open-source, so anyone can learn from, copy, and modify the design to suit their work. And it's not a toy; Asli's woven real, useful bits of fabric with it. It's also ideal for anyone who doesn't have a Victorian cloth mill in which to store it: in use, the loom takes up 600(L) × 440(W) × 310(H) mm, and folds away to 202(L) × 440(W) × 310(H) mm.



Right
Laser-cut plywood and 3D-printed attachments (plus the nous to make it work) are all you need to build your own weaving machine



REGULAR

Cold War phone Nixie clock

By **Bad Dog Designs** bad-dog-designs.co.uk

We've marvelled at the work of Paul Parry (aka Bad Dog Designs) before in these pages; here's one of his latest builds. Paul has got his hands on some Cold War-era military field telephones, and, of course, has turned them into Nixie tube clocks.

Both telephones are housed in bakelite enclosures; the lighter-coloured one (shown here) is a Russian TA-57 telephone dating from 1968–1975; the darker one is a Polish TAP-67 dating from somewhere between 1968 and 1974.

Both phones feature programmable RGB LED back-lighting to show off the Nixies in their best light, four Z570M Nixie tubes, and tell the time accurate to plus/minus one minute per year.



Right
The Nixie tubes in Paul's latest clocks should last for 20 years or more


COLD WAR
PHONE CLOCK

Planet Spinner

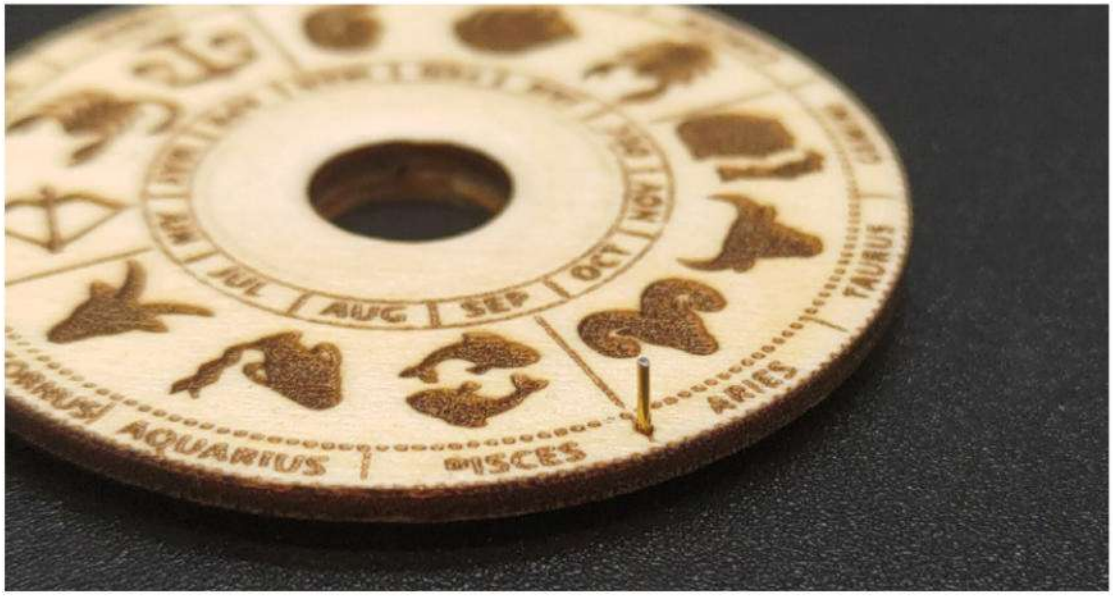
By **Frans van Hoesel** hsmag.cc/PlanetSpinner

An orrery is a mechanism that shows the relative positions of the bodies in our solar system. You can have small ones, showing just the Earth, the sun and the moon, and there are larger ones showing all nine planets (Pluto is still a planet, whatever scientists may say).

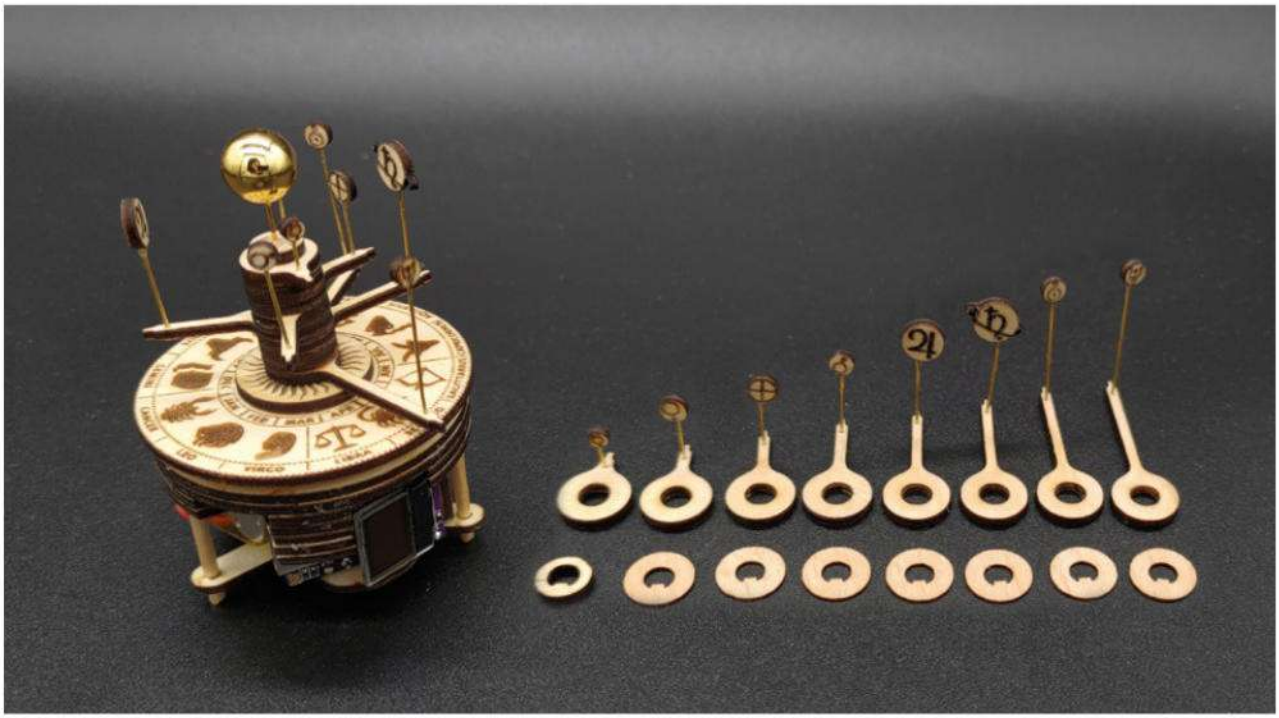
This is another example of a build that is simple on the surface – it's comprised of laser-cut plywood and toothpicks – but that's only because of the hard work and original thinking of the original creator.

There are no complex gears on this orrery. Instead, a tiny stepper motor rotates the disc that shows the signs of the zodiac. This has a pin sticking up from it, and this in turn rotates all the planets until the one closest to the sun is in place. The disc then spins back in the opposite direction until the next planet is in the right position, and so on, until the entire solar system is in place (at least, the solar system as it was known before 1781, when William Herschel discovered Uranus). Frans says that he's "surprised that no one before me came up with the idea to move planets like this without a complex gearing system." We're not surprised at all; it's a leap of imagination that makes it different from any other orrery we've seen, and yet it makes perfect sense when you see it in action.





Right
The Planet Spinner resets once a day at midnight, and takes its data over the internet from NASA



Planet
Spinner
Reset_

REGULAR

PicoCray

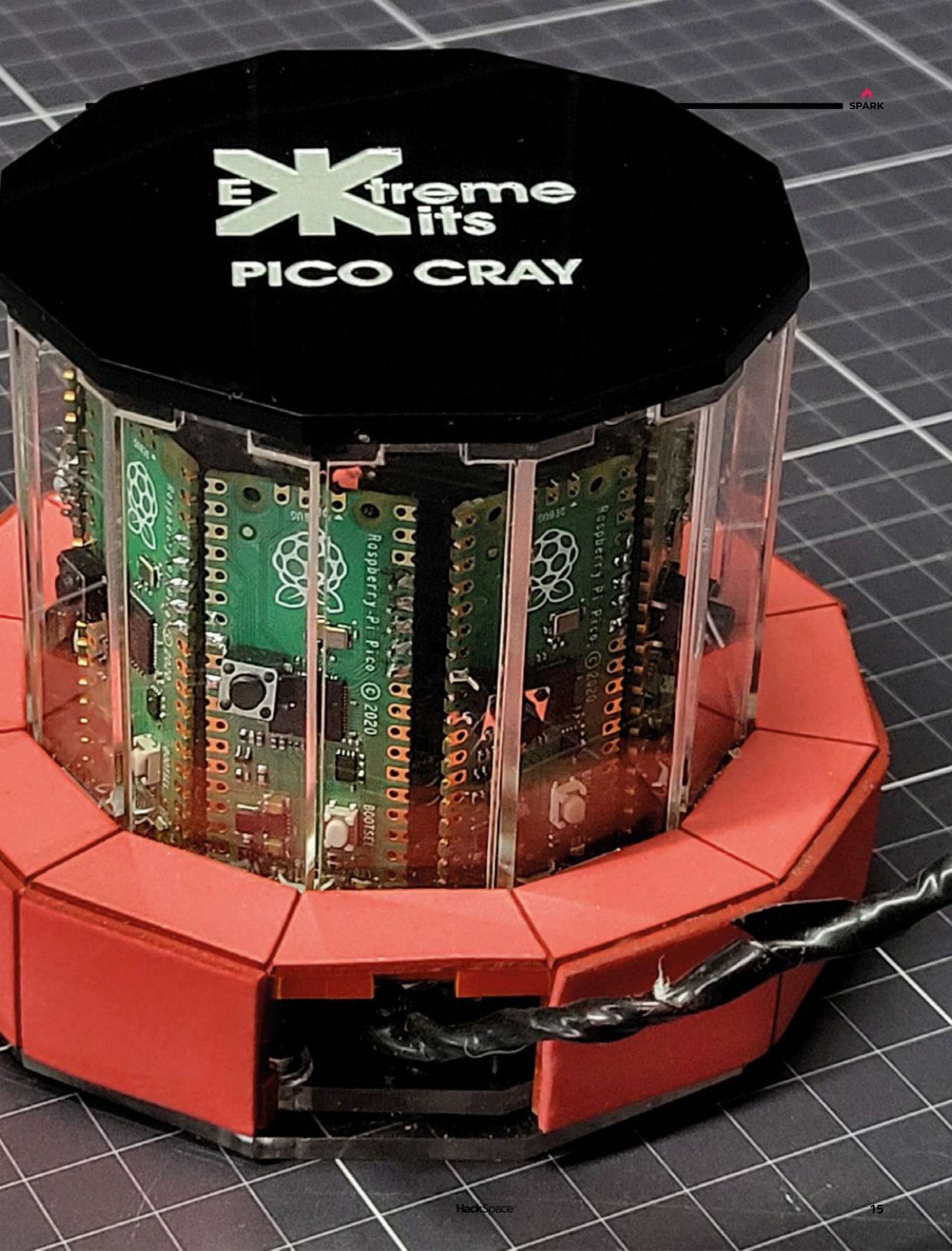
By **Derek Woodroffe** extremeelectronics.co.uk

What's the first thing that happens when new computer hardware is released? Someone runs DOOM on it. What's the second thing that happens? Someone builds a cluster. It's been over two years since the Raspberry Pi Pico came out, and in that time nobody had built a Pico cluster, so Derek Woodroffe decided that he'd do it himself.

Inspiration for the build came from the 1970s Cray supercomputers and, with a little custom electronics and the magic of I2C, Derek's put together a very small supercomputer that he's using to calculate Mandelbrot sets. If you like the sound of this, tune in next issue for more details on the hows, the whys, and the wherefores.



Right
A Cray wouldn't be a Cray without some kind of seating around it. Derek's replicated this in 2mm thick red foam


Extreme Kits
PICO CRAY
Raspberry Pi Pico © 2020
Raspberry Pi Pico © 2020

Objet 3d'art

3D-printed artwork to bring more beauty into your life



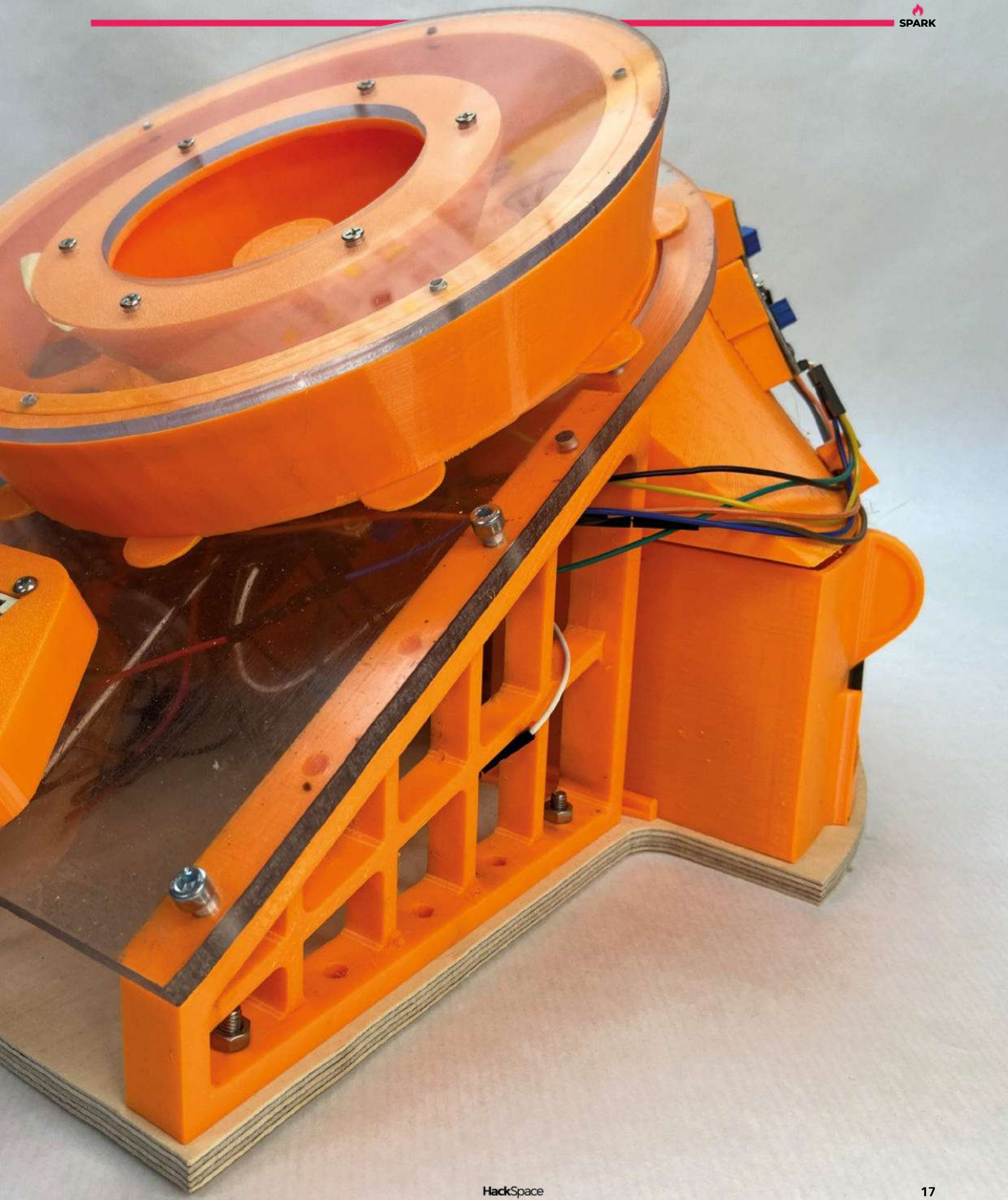
In the olden days, before we paid for things with contactless cards, we used to hand over little metal discs to shopkeepers in exchange for goods. Unbelievable, right? What's more, everything used to cost really weird numbers, like £9.99, so you'd hand over your money and get a tiny bit of money back, which used to rattle around uselessly once you got home.

Seriously though, we still use coins, and we still have jam jars full of old bits of shrapnel that we really should convert into more useful money. That's where this beautiful bit of design by Fraens comes in. His coin sorter uses a rotating drum with differently sized holes in it to sort coins in order from smallest (10c) to largest (€2).

There's a TRCT5000 infrared sensor mounted on each coin ejection slot, to detect when a coin falls through, and an Arduino to count each coin and keep a tally of how much your change is worth.

Other than 3D-printed parts, the machine has a laser-cut plywood base and a clear acrylic lid, so you can watch your money spinning around and falling like a metaphor too complicated for us to understand.

hsmag.cc/CoinSorter



Letters

ATTENTION ALL MAKERS!

If you have something you'd like to get off your chest (or even throw a word of praise in our direction) let us know at **hsmag.cc/hello**

LET US BUILD

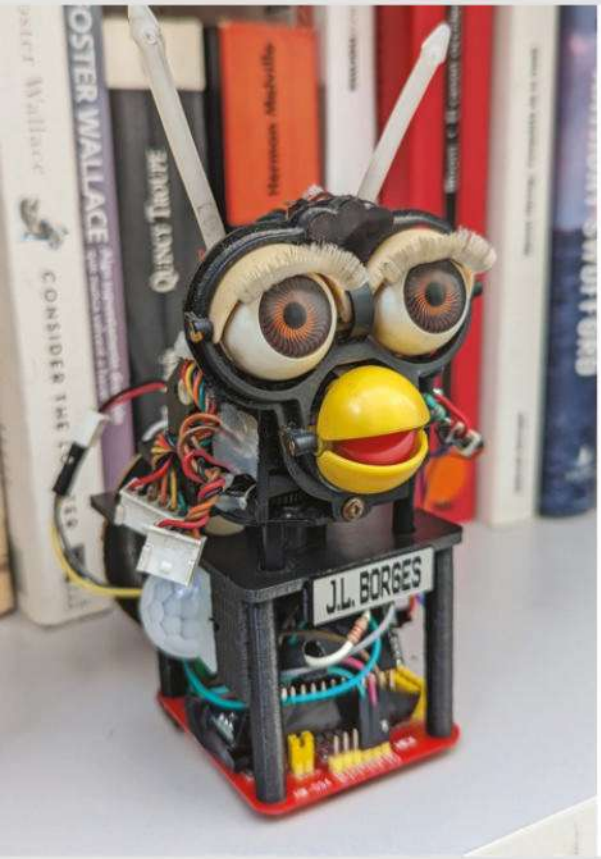
It feels like we're on the edge of a transport revolution. Electric motors are (fairly) cheap and (fairly) easy to work with at the size needed for modest personal transportation. They're even relatively easy to retrofit onto human-powered transport, as Andrew Lewis showed in his e-scooter build in issue 67. However, politicians are dragging their feet on the legislation needed to make them legal. Perhaps we need an orchestrated maker campaign to get Westminster to let us build our own vehicles. We could have a future of personalised machines made specifically for each person's needs. Want small and lightweight? No problem. Need to move the kids? Sure, just bolt on a few extra seats. Delivering cargo? You decide how big to make your carrier.

Theo
Bristol



Ben says: We try not to be too political here at HackSpace magazine, but it's utterly absurd that, here in the UK, it's legal to rent an electric scooter from a multinational corporation but not to buy or make one yourself. Obviously, there will need to be some regulation to ensure people build machines that are not excessively dangerous, but there's a long history of homemade bikes, so this doesn't feel like it should be a difficult hurdle to jump.

Rest assured that when politicians do finally come to their senses (we're not holding our breath), we'll have a full series on how to create your own *Mad Max*-inspired cyberpunk chariot.



MECHANICAL MONSTROSITIES

Can you please not feature any more Furby projects? They are creepy beyond words and should be cast aside. It was a mistake to ever manufacture them. We should, as a species, leave them behind to rot in the landfills of the world where they belong.

Robin
Dover

Ben says: There is occasionally a fine line between creepy and cute, and Furbys sit directly on this line. Some parts of them are quite lifelike, some completely mechanical. I think part of the reverence for them today is that we don't seem to make mechanical things like this any more. Were they made today, they'd have miniature screens to display their faces, not animatronic eyes.

They also originate from a time when artificial intelligence (AI) was just starting to be something that people were aware was a real thing. Though it seems absurd now, an internal memo at the US's National Security Agency (NSA) read: "We are prohibited from introducing [Furbys] into NSA spaces. Those who have should contact their Staff Security Office for guidance."

As we're living through another wave of AI panic, it's perhaps reassuring that the Furbys haven't (yet) taken over the world.

We can't quite put our fingers on it, but Furbys say something about what it means to be human in an increasingly machine-dominated world. They span the gap between animal and machine (at least in our minds). Yes, they sometimes feel uncomfortable, but in that discomfort is a message – a lesson on how we should build the future. We're still trying to fully understand that lesson, and, until we do so, we'll continue to feature Furby projects. Sorry.

THANKS JO

It's hard to think of someone who has done more for hobbyist 3D printing than Josef Průša, other than perhaps Adrian Bowyer. Not only has he created an iconic series of machines, but he has also kept his designs open-source for others to learn from.

Along with Limor 'LadyAda' Fried, he's an inspiration showing how you can build a successful business on open-source hardware. I don't know where the maker scene would be without people like these.

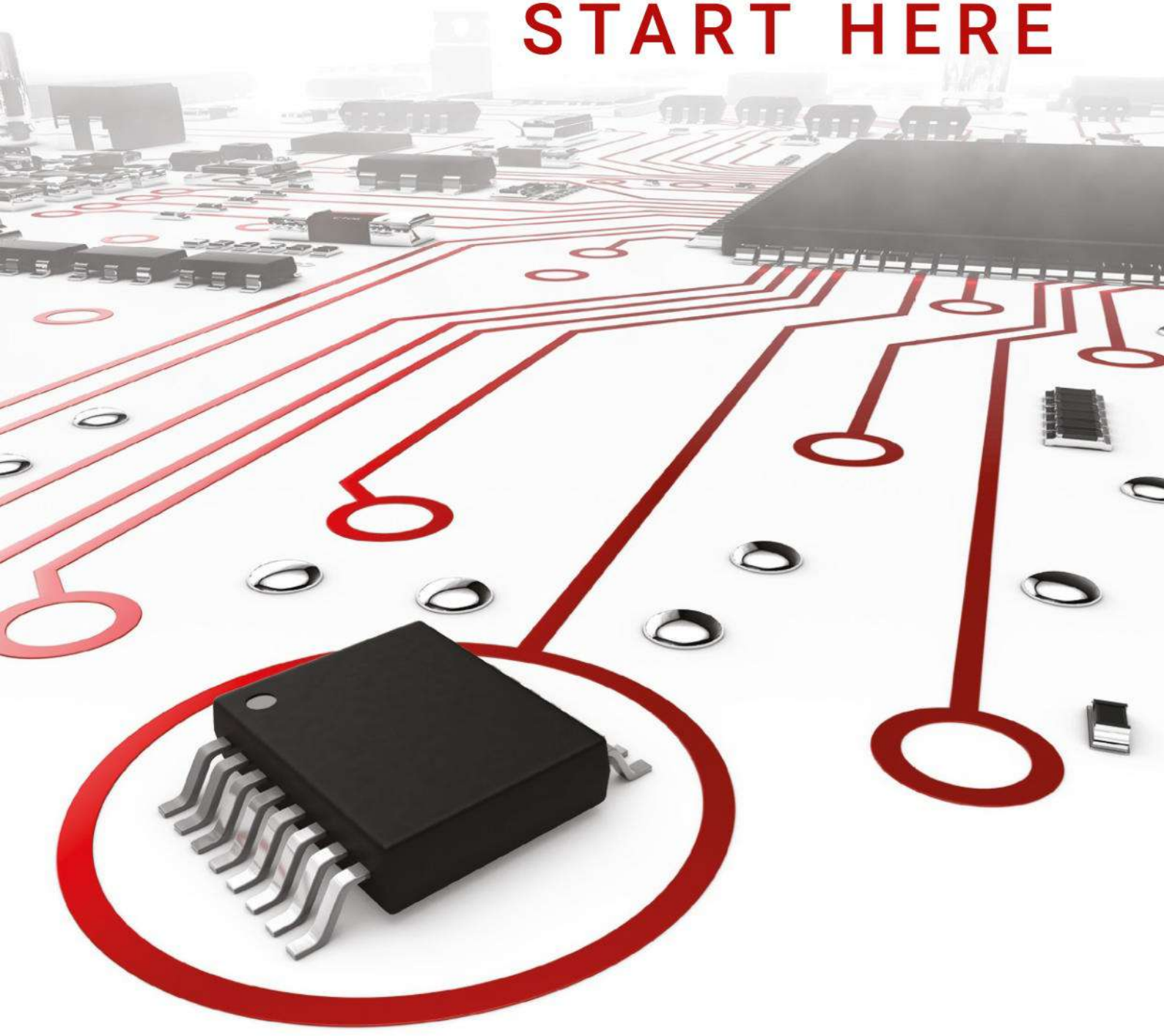
Alexis
Norwich

Ben says: The thing that unites Prusa and Adafruit hardware is how easy it is to work with. It's well-documented, and there are people to help when things don't quite seem right. It's also hackable and easy to tweak it to your own needs. In both cases, you can get cheaper hardware (and in some instances, that might be a better option), but for the extra money, you get a whole lot of value.



NEW PRODUCTS

START HERE



Looking for the latest products? Look no further. With over 400,000 new products stocked, we've got your electronic component and automation needs covered.

Find it at **digikey.co.uk/new** or call 0800 587 0991.

Digi-Key ELECTRONICS

Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2023 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER
Supporting The Authorized Channel

HackSpace
TECHNOLOGY IN YOUR HANDS

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

PG 32

HOW I MADE:
OFF-GRID POWER TOOLS

Go camping, and take your workshop with you

PG 38



INTERVIEW:
TIM HUNKIN

We meet the man who inspired our childhoods

PG 46

IN THE WORKSHOP

A lot of LEDs and not enough pins

PG 22

DIY SMART CAMERA

CAPTURE IMAGES YOUR OWN WAY

FEATURE



DIY SMART CAMERA

WITH A RASPBERRY PI AND CAMERA MODULE, THE POSSIBILITIES ARE NEAR ENDLESS





Phil King

A long-time Raspberry Pi user and tinkerer, Phil is a freelance writer and editor with a focus on technology.

Combining the power and versatility of a Raspberry Pi single-board computer with a compact Camera Module – or larger HQ Camera with interchangeable lenses – enables you to take great pictures and videos and do a whole lot more. There's also the option of 3D-printing or modifying a case to create an all-in-one compact camera. Or, one with an eyepiece tube to fit a telescope for spectacular shots of the night sky.

Using the NoIR Camera Module variant, which omits the IR filter, you can even capture images in the dark (with IR LED illumination) – ideal for night vision, nocturnal creature photos, or even a ghost detector.

With the addition of machine learning computer vision, advanced projects enable you to identify objects and monitor your body movements during workouts.

Other fun project possibilities include a photo booth, security surveillance system, wildlife camera trap, and time-lapse photography. Lights, camera, action!...

MAKE A TIME-LAPSE VIDEO

COMBINE A SERIES OF STILL IMAGES INTO A SPED-UP VIDEO

One of the easiest, and yet most impressive, uses for a Raspberry Pi camera is to film a time-lapse video created from a series of still images.

Sometimes used for interludes in film and TV dramas, to indicate the passing of time between scenes, a time-lapse video is an impressive and versatile effect that appears to speed up time. Classic examples include clouds passing across the skyline, city traffic moving (a lot faster than in reality), star trails, a sunset or sunrise, and plants blooming/growing.

With a Raspberry Pi Camera Module or High Quality Camera, you can easily create effective time-lapses. You just need to set everything up correctly and make sure the camera doesn't move during the filming period.

STEP 1: CONNECT THE CAMERA

You will need to connect the Camera Module or High Quality Camera before switching on your Raspberry Pi, otherwise it won't be detected. If the ribbon cable isn't already connected to the camera's CSI connector, pull the plastic tab out and insert the cable (with the covered blue side facing the plastic tab), then push the tab back in. On the Raspberry Pi, insert the other end of the cable into its Camera port in similar fashion. Note that on a Raspberry Pi Zero, the Camera port is on the edge of the board and is narrower, so requires a special ribbon cable/adapter. Make sure both cable ends are snug and secure – a poor connection can stop the camera working.

STEP 2: REMOTE ACCESS

You can connect the Raspberry Pi to a monitor to check the camera is working – by entering the `libcamera-hello` terminal command for a five-second camera preview.

When filming a subject, however, you may well want to position your camera away from a monitor – in which case, you will need to access the Raspberry Pi remotely via SSH (Secure Shell) to control it. You can set up SSH in the Raspberry Pi Configuration menu's Interfaces tab, or in the raspi-config tool from the command line.



Above
Make sure that the camera is connected to the correct port

> **You can connect the Raspberry Pi to a monitor to** check the camera is working

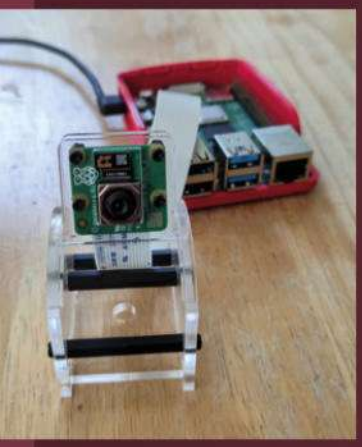
Alternatively, you can enable SSH when writing the Raspberry Pi OS image to a microSD card in the advanced options of the Raspberry Pi Imager tool (click the cog icon after choosing an OS), as well as altering the host name and/or setting up the Wi-Fi connection to your router so the Raspberry Pi automatically connects to it on first boot.

If you're only going to be using the Raspberry Pi remotely, you may want to just install the Lite version of Raspberry Pi OS, which lacks a desktop and therefore takes up less room on the card and is quicker to boot.

You can SSH into the Raspberry Pi from the terminal/PowerShell of another computer using the Raspberry Pi's IP address:

```
ssh [username]@[IP address]
```

You can find the IP address of your Raspberry Pi by entering `hostname -I` at the command line – or, →



Above
If you don't have a purpose-made camera mount, you can make one

if you don't have it connected to a monitor, by checking the devices list on your wireless router web page to find it.

Alternatively, you can use the Raspberry Pi's host name instead of an IP address. It's **raspberrypi.local** by default, but can be set to another name in Imager's advanced options.

```
ssh [username]@raspberrypi.local
```

Either way, you'll be prompted to accept the connection by typing 'yes' (the first time), then asked to enter the Raspberry Pi's password. You'll then see the usual command line prompt for it.

STEP 3: POSITION YOUR CAMERA

The main thing to ensure when filming a time-lapse video is that your camera doesn't move, otherwise it'll ruin the effect. So it's best to place it in one of the many Camera Module mounts, which can be bought cheaply – or, if using an HQ Camera, you can screw it onto a standard camera tripod.

An alternative for filming exterior scenes is to stick the camera to a window using a special mount with suction cups, such as the ZeroView (**hsmag.cc/ZeroView**) which can also hold a Raspberry Pi Zero W on the rear.

At this point, you may want to take another test shot to check the focus and light level for the scene. The V1 and V2 Camera Modules have fixed focus, while the V3 has the option of continuous autofocus. The exposure mode is automatic by default, although you can use the `--ev` parameter to make it lighter/darker or use one of several exposure scenes, e.g. night, with the `--ex` parameter. Note that conditions may change during the shoot, however.

The HQ Camera's focus and aperture can be adjusted manually by rotating parts of the attached C- or CS-mount lens (see **hsmag.cc/HQCmountLens** or **hsmag.cc/HQCSmountLens**).

STEP 4: TAKE A TEST SHOT

The latest versions of Raspberry Pi OS have libcamera-apps pre-installed. While these replace the previous `raspistill` and `raspivid` commands, they work almost the same way with a similar range of options. To take a test still image with the camera, enter:

```
libcamera-still -o test.jpg
```

You'll see a stream of information in the terminal as the shot is taken. If you're using a desktop version of Raspberry Pi OS and are connected to a monitor (or are accessing the desktop remotely via VNC), you can view the photo in the Image Viewer desktop application. If not, you can copy it to your local computer using SCP (Secure Copy):

```
scp [username]@[Pi's IP or hostname]:test.jpg
 [/local/path/to/save]
```

Enter the Raspberry Pi's password to proceed. Check that the test shot looks OK before proceeding with filming the time-lapse. Otherwise you may want to reposition the camera or alter some of libcamera's settings – see the documentation for more info: **hsmag.cc/libcamera**.

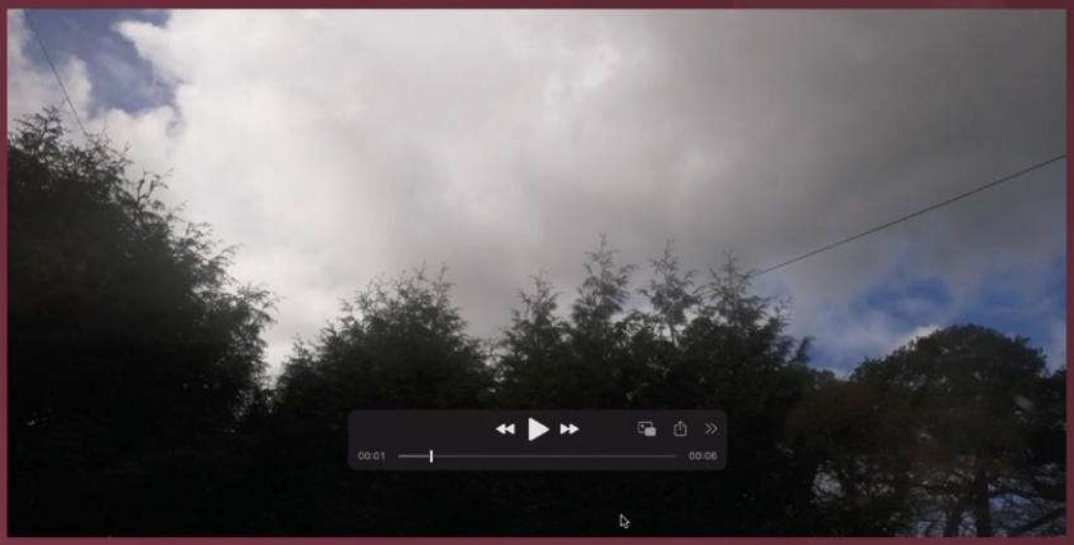


Above
Summer has finally come!

RASPBERRY PI CAMERAS

There is a range of official Raspberry Pi cameras available. For more details and full specs, see **hsmag.cc/RPiCameraDoc**. Third-party cameras are also sold, but may require their own software libraries.

- **CAMERA MODULE V2:** An upgrade of the original 5MP, fixed-focus Camera Module, this tiny camera features adjustable focus and an 8MP sensor. Available in standard and NoIR (for low-light shooting) versions.
- **CAMERA MODULE V3:** Upgraded with HDR (high dynamic range), a 12MP sensor, and – most notably – motorised focus which can adjust automatically when shooting video. Available with a standard (70°) or wide-angle (120°) lens, both coming in standard or NoIR versions. It can also record HD video at 50fps.
- **HIGH QUALITY CAMERA:** A larger board with a 12.3MP sensor, it enables you to attach different C- and CS-mount lenses with manual focus and aperture. There is also a version for M12 lenses.
- **GLOBAL SHUTTER CAMERA:** Similar to the0 HQ Camera, with interchangeable lenses, it has a much lower 1.6MP resolution, but a global – not rolling – shutter that's ideal for shooting fast-moving subjects.



Left
We love watching clouds move in fast-forward

STEP 5: LET'S DO THE TIME-LAPSE

You can start shooting a time-lapse with a single `libcamera-still` command with the `--timelapse` option and a few parameters. For example:

```
libcamera-still -t 30000 --timelapse 2000 -o
 image%04d.jpg
```

Here, the number after `-t` is the total duration in milliseconds, in this case, 30 seconds. The number after `--timelapse` is the interval between images in milliseconds, in this case, 2 seconds. Finally, we use `%04d` to add a four-digit number to the file name, after 'image', starting with '0000'.

By varying the timings, you can take much longer time-lapses with more or less frequent image captures. For slower scenes, such as clouds and star trails, use a longer interval between captures; for faster ones, such as traffic or people, a shorter interval is better to capture the movement without it appearing too disjointed. If you're shooting a long duration via SSH, you may want to use the `nohup` prefix before the command. Short for 'no hangup', this will ensure it continues working even if the SSH connection is lost.

STEP 6: COMPILE THE VIDEO

Once you have your sequence of stills, it's time to compile them into a time-lapse video. For this, a good option is the ffmpeg command-line tool. It should be pre-installed in the latest version of Raspberry Pi OS; if not, install it with:

```
sudo apt install ffmpeg
```

Compile the time-lapse video from your images with a command like this:

```
ffmpeg -r 10 -f image2 -pattern_type glob -i
 'image*.jpg' -s 1280x720 -vcodec libx264
 timelapse.mp4
```

Here, the number after `-r` is the frame rate, in this case, 10 fps – try varying it for different results. The next part tells it to pattern-match your set of image files using the * wildcard. The numbers following `-s` (scale) determine the video resolution. The final part uses the x264 encoder to save the video as an MP4 file.

Depending on the Raspberry Pi model and the number of images, this may take a while to complete. A Raspberry Pi 4 is pretty quick, but other models are slower (a Raspberry Pi 3 will encode around two frames per second). So you may instead want to transfer the image files to a more powerful computer via SCP:

```
scp [username]@[Pi's IP or hostname]:image\*.
 jpg [/local/path/to/save]
```

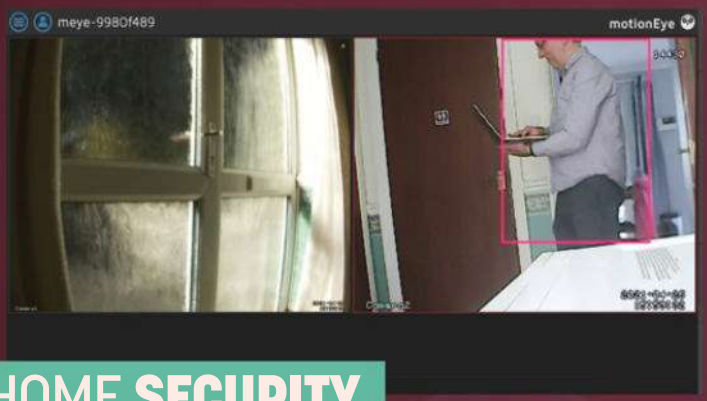
...and then use ffmpeg on that to compile the video much faster. Now you know the basics, have fun experimenting with different subjects and interval/frame rates to create impressive time-lapse videos. →

OCTOLAPSE

As well as using OctoPrint on a Raspberry Pi to monitor and control your 3D printer via a web interface, you can install a plug-in called Octolapse that enables you to shoot cool 3D printing time-lapses. See **hsmag.cc/Octolapse** for more details.

RASPBERRY PI CAMERA PROJECTS

INSPIRATIONAL IDEAS FOR YOUR RASPBERRY PI CAMERA MODULE



HOME SECURITY

Low-cost and tiny, Raspberry Pi Camera Modules are ideal for home security video surveillance. There's even a special operating system, motionEyeOS (**hsmag.cc/motionEyeOS**), that makes it super-easy to set up one or more cameras which can be viewed and controlled via a web interface.

Above
The rectangle indicates that motion has been detected (by comparing frames)

Just head over to the GitHub repo's supported devices page (**hsmag.cc/motionEyeOSDevices**) and download the relevant version for your Raspberry Pi model (or other single-board computer). The wiki has full instructions for how to configure everything, including a network of multiple cameras/devices – including USB webcams. Numerous features include motion detection to trigger photos/videos, email/push alerts, and time-lapse videos.

You could use it to catch porch package thieves. Alternatively, check out Canadian maker Ryder's custom deterrent system that uses machine learning to recognise packages and people, triggering an alarm and showering the thief with flour and water (**hsmag.cc/ThiefDeterrent**).

WILDLIFE CAMERA

A popular use for a Raspberry Pi Camera Module is for capturing close-up shots of wildlife. By adding a PIR sensor to detect movement, you can activate the camera whenever animals are in the field of view – although you may also get a few false positives triggered by people or vehicles passing by.

When placing your camera outdoors, you'll want to protect it and the Raspberry Pi from the elements with a weather-proof enclosure. This could be a transparent box or something more purpose-built, such as the NatureBytes (**naturebytes.org**) Wildlife Camera Case – there's also a fully-fledged kit, including a Raspberry Pi and rechargeable battery pack. Just point it at a bird feeder and you'll get some spectacular shots of your feathery friends.

To take photos of nocturnal creatures, you could set up a similar PIR-triggered camera trap at ground level with a NoIR Camera Module and an IR LED ring (**lisiparoi.co.uk**) that fits around its lens for non-visible flash/illumination. Alternatively, you could place a NoIR camera in an empty bird box, ready to view its occupiers when they arrive: **hsmag.cc/IRBirdBox**.



Above
Pointing your camera at a bird feeder is a sure-fire way to photograph various species, such as these long-tailed tits

COMPACT CAMERA

While the Raspberry Pi Camera Modules can be used in mounts or, in the case of the HQ Camera, on a tripod, you can also incorporate one into a case so it looks more like a traditional compact camera.

Among the numerous 3D-printable designs available, one project that caught our eye is PolaPi Zero: **hsmag.cc/PolaPiZero**. Not only does its case enclose a Raspberry Pi Zero and Camera Module, but there's an LCD on the rear for a live view. Inspired by old-school Polaroid cameras, it also features an Adafruit Nano thermal printer that produces a physical monochrome copy of your photo.

If you want to use the larger HQ Camera on the move, check out James Martel's project (**hsmag.cc/RPi4HQCamera**) for inspiration. He modified an existing case for a Raspberry Pi 4 and LCD to incorporate a mounting plate for the HQ Camera. Alternatively, you could hack an official Raspberry Pi case to add an HQ Camera and screen, like Richard Hayler did – see The MagPi issue 94 for a guide: **hsmag.cc/MagPi94**.



Above
Richard Hayler's modified case packs in an HQ Camera, Raspberry Pi 4, and LCD

PHOTO BOOTH



Above
For illumination, this booth features a strip of ultra-bright LEDs diffused by an angled piece of wood.
Credit: Jonathan Lang Photography

A fairly simple Raspberry Pi camera project is to build your own photo booth, which is a fun addition to any event. Baulking at the high hire prices for commercial booths, Jack Barker made one for his own wedding (to Pam). Building the main booth structure from plywood, he cut out a section for an LCD screen and drilled holes for a large arcade button below and the Raspberry Pi Camera Module above.

Pressing the button starts a seven-second countdown with an instructional slide, followed by a live view from the camera so you can see yourself on screen and perfect your pose.

While the wedding venue's lack of Wi-Fi scuppered Jack's plan for live online photo backup, he added a printer to the booth to produce instant copies of the photos.

If you fancy making your own photo booth, take a look at Jack's seven-part build guide: **hsmag.cc/JackPhotoBooth**. →

NIGHT VISION

All of the standard Raspberry Pi Camera Modules are available in a NoIR version which omits the infrared filter. This gives it the ability to see in the dark with infrared lighting, opening up all sorts of possibilities, such as monitoring nocturnal animals and the health of plants. Note that it is also possible to manually remove the IR filter from standard Camera Modules and the HQ and GS Cameras – see **hsmag.cc/RPiCameraDoc** for details.

Dan Aldred built a night vision camera (**hsmag.cc/NightVisionCam**) enabling him to see in the dark, just like in the Splinter Cell classic video game. While he used a third-party camera with IR LEDs, connected to a Raspberry Pi Zero, you could instead use a standard NoIR Camera Module to creep around like a stealth operative.

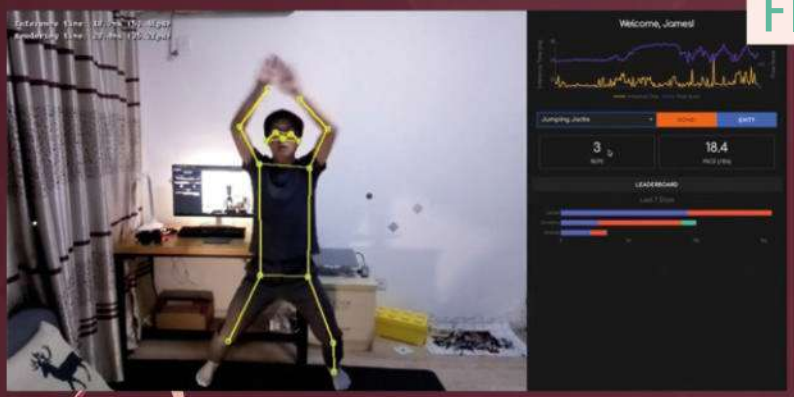
Or you might want to try detecting some ghosts, like Anthony DiPilato. Not only does his handheld Ghost Detector (**hsmag.cc/GhostDetector**) incorporate a NoIR Camera Module and touchscreen for night vision, but also EMF sensors with twin antennae, mic, compass, temperature sensor, and even a Geiger counter.



Above
See in the dark with a night vision camera, perfect for a midnight feast

Left
I ain't 'fraid of no ghost! Scan for paranormal presences with this Ghost Detector

FITNESS TRAINER



Above
HIIT Pi: each video frame is analysed using the PoseNet model in TensorFlow Lite to detect and track the user's pose

A positively healthy use for a Raspberry Pi camera is to monitor your body movements while you're exercising. With the use of computer vision, you can then analyse how well you are doing during your workouts.

This is especially important if you're engaging in high-intensity interval training (HIIT) on your own, to make sure you're doing it right for optimum results. Using a Raspberry Pi aided by Google Coral USB Accelerator, James Wong created a machine learning Raspberry web app to track his movements and poses and then score them based on a set of pre-defined standards. Find code and instructions on his GitHub repo: **hsmag.cc/HIITPi**.

If you prefer your exercise to be a little less intense, Salma Mayorquin and Terry Rodriguez's YogAI (**hsmag.cc/YogAI**) combines a Raspberry Pi magic mirror with AI yoga pose recognition to help guide you.



ASTROPHOTOGRAPHY

Due to its compact size, a Raspberry Pi Camera Module is ideal for using with a telescope to take images of the night sky. David Palmer's PiPiece project (**hsmag.cc/PiPiece**) is based around a 3D-printable case for a Raspberry Pi Model A and Camera Module with an eyepiece tube design available in various sizes to fit most amateur astronomy telescopes.

The project does involve removing the tiny lens of the Camera Module, which should be done carefully to avoid damage to the sensor. Once it's fitted to the telescope, you can adjust the latter's eyepiece to focus, as you would normally. The field of view is small, so fine alignment of the telescope is required to observe a planet or other celestial body.

Alternatively, if you lack a telescope, a Raspberry Pi HQ Camera fitted with a suitable zoom lens should enable you to take detailed shots of the moon – it requires good aperture selection and careful focus. For more on astrophotography, check out the feature in The MagPi issue 128 (**hsmag.cc/MagPi128**).

Above
The PiPiece attached to a telescope, enabling the Camera Module to capture the night sky

VISION AID

For blind and partially sighted people, a Raspberry Pi and camera can provide a way of seeing by analysing the view and turning it into audio. Inspired by a blind cousin, Robert H'obbes' Zakon built a Seeing Wand that can speak the name of whatever it's pointed at. When the user presses the button on the prototype device, the Camera Module attached to its Raspberry Pi Zero takes a photo; this is sent to Microsoft's Cognitive Services Computer Vision API to obtain a description, which is then spoken using the eSpeak text-to-speech tool. Find the build details at **hsmag.cc/SeeingWand**.

Alternatively, the All-Seeing Pi (**hsmag.cc/AllSeeingPi**) is worn on a partially sighted user's head and sends a video feed from the camera to a display right in front of their eyes to give them a more detailed, close-up view of the environment.



Above
It may look a little makeshift, but the Seeing Wand is easy to build and the code is on GitHub: hsmag.cc/SeeingWandGH

SUBSCRIBE TODAY

FOR JUST £10



Get three issues plus a
FREE Raspberry Pi Pico W
delivered to your door
(UK only)

hsmag.cc/FreePico

Subscription will continue quarterly unless cancelled

NEW INTERNATIONAL OFFERS!



Get 6 issues plus a **FREE Raspberry Pi Pico W** for just...

$43 USA / **€43** EU

£40 Rest of the world

Head to **hsmag.cc/subscribe** to get yours today!



FEATURE

HOW I MADE

By Andrew Lewis

POWER IN THE WILDERNESS

Charging your power tool batteries off-grid

Working with power tools in the wilderness can be fun, whether you're building a secret bunker in expectation of societal collapse, or repairing your bike at the side of the road. Unfortunately, you're limited in how much you can do by the need for power. Solar and 12V engine power can't be used to charge power tool batteries directly, and not every manufacturer provides a DC power solution for charging their products. In this article, you'll see how to use some off-the-shelf components to create your own power tool charger that will run off anything from a cigarette lighter socket to a solar cell.

SAFETY

LiPo batteries pack a lot of power into a small space, and can discarge that power quckly. This is both the best and worst thing about them. When controlled, it means we can power all sorts of things without taking up too much space. When uncontrolled it means we have an explosion.

In order to charge LiPo batteries safely, you have to understand how LiPo batteries work in general, how your speciffic LiPo batteries work, and how your specific charger works.

If you don't feel confident with all three of these, then it's probably best to work with off-the-shelf solutions from brands you trust until you gain the expience and knowledge to do this safely.



Left
If you're off-grid, it makes sense to cut out the mains inverter whenever you can. This charger has been converted to run from 12V DC

What I used

- 5S battery management system **hsmag.cc/5S-BMS**
- DC buck-boost converter **hsmag.cc/DCBuckBoost**
- Arduino Uno **hsmag.cc/UnoRev3**
- ACS712 current sensor **hsmag.cc/ACS712Module**

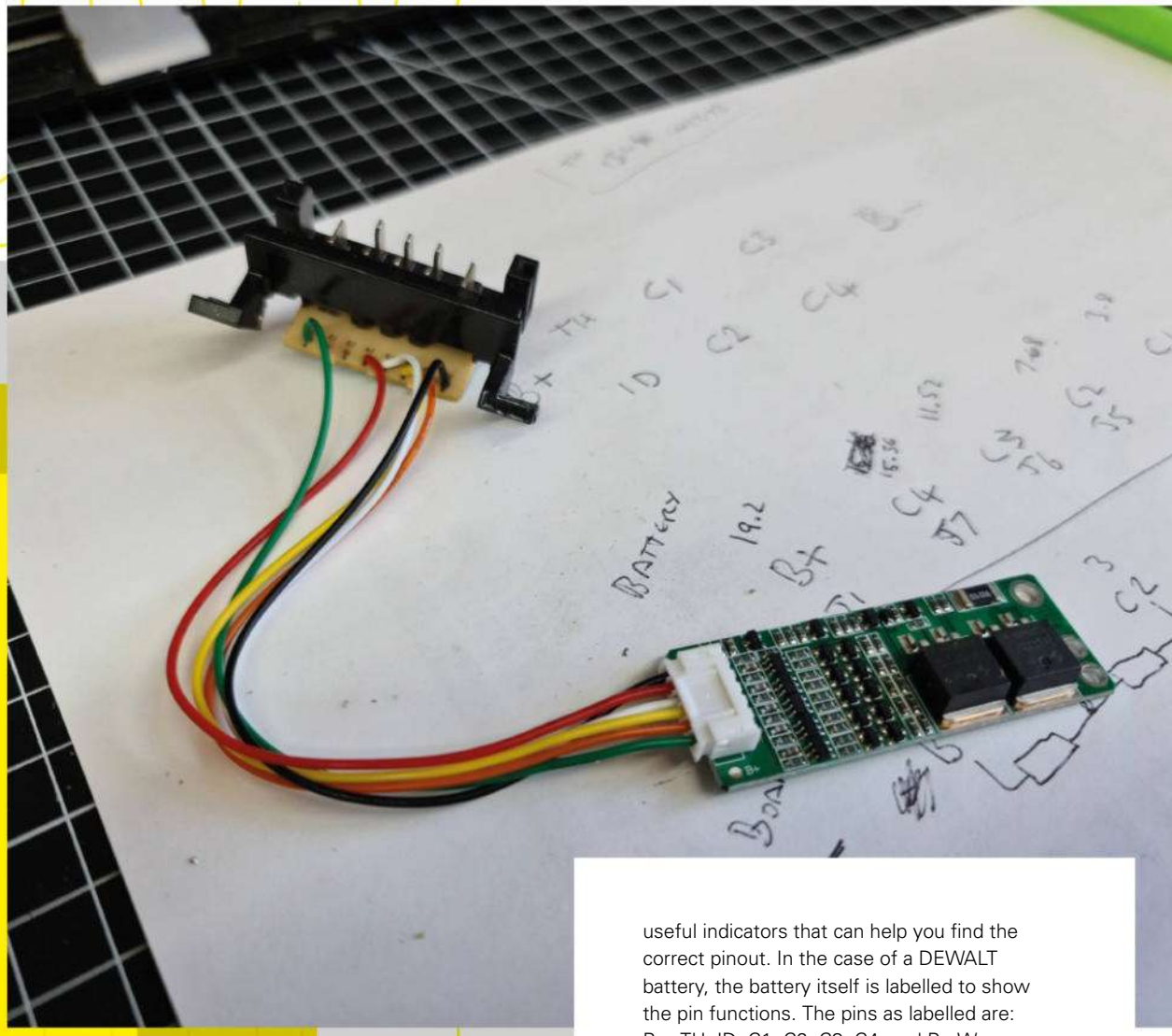
Modern power tool batteries can be dangerous things. We've all seen YouTube videos of LiPo batteries erupting like the Fagradalsfjall volcano in Grindavik after they've been mistreated, and that volatility makes the idea of messing with them a little bit scary. You should have a healthy amount of fear of LiPo batteries, because they are dangerous, but that doesn't mean that we can't work with them. It just means that we have to be careful. In this article, we'll look at one way of charging them, but only undertake this if you properly understand the risks involved.

The first thing you need to do is look at the battery and charger, taking note of any



Above
This DEWALT battery has handy indicators for each pin to tell you what they do. Take care when translating these pins to a BMS, as the pins on the PCB can be in a different order. Take special note of the locations of the positive and negative terminals relative to pins C1 to C4

FEATURE

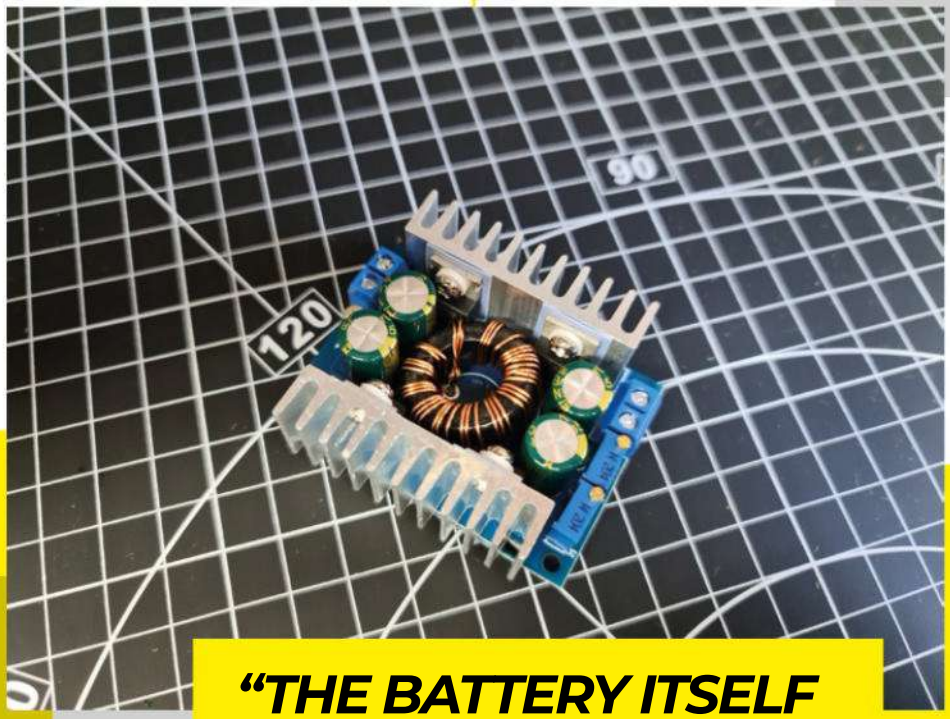


Above
The BMS wired up to the connector for the battery. The notes used to figure out the pinout can be seen on the sheet of paper below. Keeping notes like this is really important to ensure you don't get confused and make a mistake when working on boards with more than a couple of connections

useful indicators that can help you find the correct pinout. In the case of a DEWALT battery, the battery itself is labelled to show the pin functions. The pins as labelled are: B+, TH, ID, C1, C2, C3, C4, and B-. We can take a good guess at what we think these pins do, and then confirm by checking with a multimeter, or even pulling the battery apart to take a look. B+ and B- have double height connections, which shows that we can expect high current from these pins – the multimeter confirms these are the main battery terminals. C1 to C4 seem likely to be connections to the individual cells of the battery, and by using the multimeter to check from the B-, we see that each of the connections C1 to C4 have an increasing voltage from around 3.8V up to 15.3V, increasing by about 3.8V for each connection. That seems to confirm that

LIMITING CURRENT

The best choice of DC-DC power supply for this project is a DC buck-boost supply that can provide a constant current of around 6A, with current limiting. Set the voltage output for the supply first, by connecting the input power to a bench power supply or battery, and checking the output voltage with a multimeter. Use the voltage-adjust capacitor to set the correct output voltage (21V for the BMS), keeping your eye on the voltage as you turn the screw on the voltage capacitor to make sure you're going in the right direction. Next, set the current to the minimum by turning the current limiting capacitor to its minimum value (this usually means turning the screw counter-clockwise at least 30 revolutions). Connect a load to the output of the power supply and look at the output voltage. If you don't have a suitable dummy load, you can use the BMS controller with a battery connected to it. Turn the load capacitor clockwise and measure the current on the input of the buck converter until the current drain is around 6A. Setting the current higher than 6A is possible if you use active cooling, but is not advisable if you aren't. The BMS runs at 21V, so 6A or just over should yield roughly 3A charging current, which is similar to the original battery charger.



"THE BATTERY ITSELF IS LABELLED TO SHOW THE PIN FUNCTIONS"

each of these pins is connected to a separate cell.

The DEWALT charger takes 240V AC, and outputs the correct voltage for charging either 12V, 14V, or 18V batteries. If we take a look at the board, we can see that it's pretty complicated. While it is possible to hack this board directly and get it to charge from a DC power source, it's actually a bit easier if we just throw the charger circuitry away and start again with some off-the-shelf components. We can simplify things more if we just assume that we are going to be using only one type of battery: in this case, the 18V flavour. The only part of the charger we need is the plastic case and the connector that attaches to the battery. You can go ahead and desolder this connector and fling everything else into the 'scavenge for parts' box in your workspace.

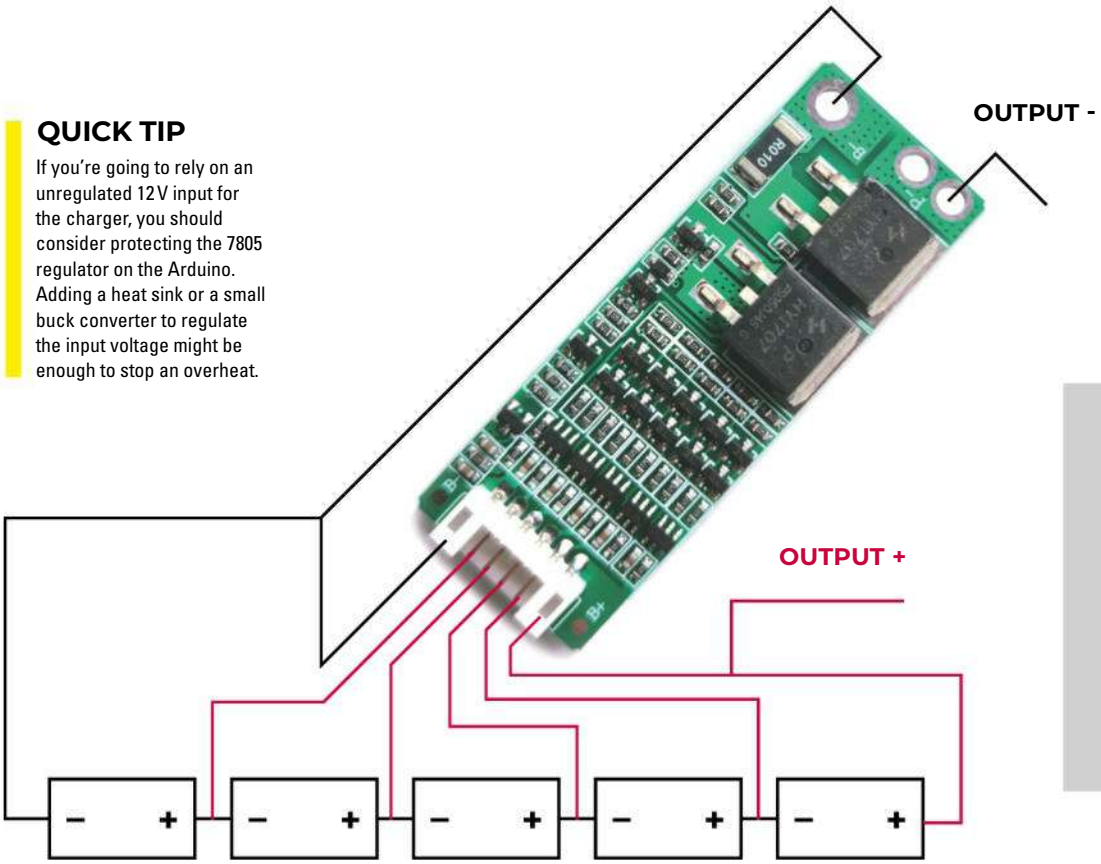
Charging a series of 18650 cells is a common problem, and there are off-the-shelf solutions for building new battery packs and chargers safely. Battery management system (BMS) boards are cheap and cheerful ready-made solutions that will protect your battery from overcharging, excessive discharge, and many other things. Where cheap BMS boards fall down is that they don't usually have any sort of indicator lights, and they don't often have any thermal protection. That's not a big problem, and you can solve it by adding an Arduino into the mix to monitor the BMS board.

Above
This power supply has current limiting and variable voltage capacitors, and also has a surface-mount fuse built directly onto the board. Adequate fuses are critical in a system like this, where unfused short circuits can be catastrophic

FEATURE

QUICK TIP

If you're going to rely on an unregulated 12V input for the charger, you should consider protecting the 7805 regulator on the Arduino. Adding a heat sink or a small buck converter to regulate the input voltage might be enough to stop an overheat.





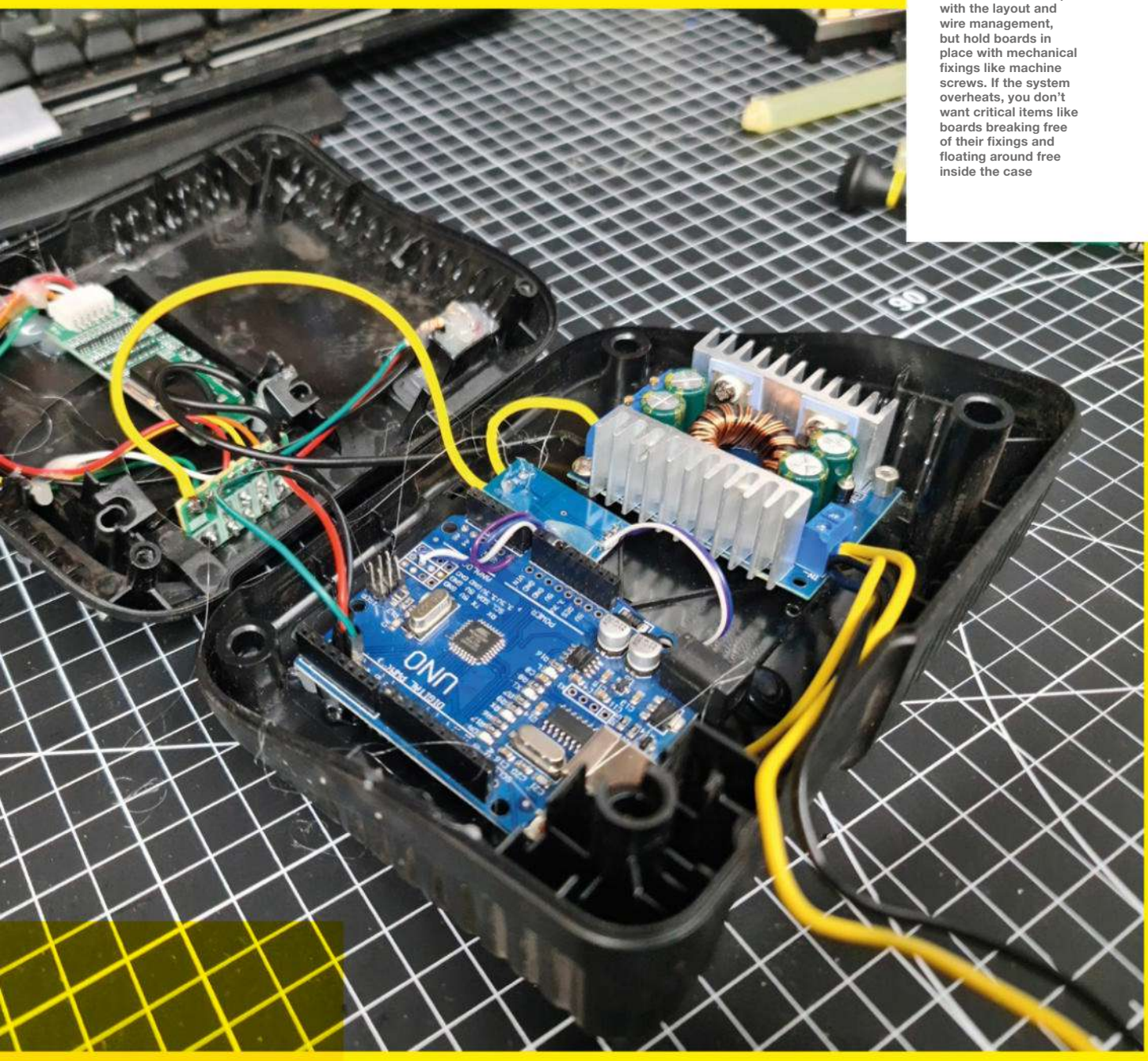
To charge an 18V battery pack, the BMS board will need around 21V. Most cars and portable solar solutions work at 12V, so we need to increase the voltage as efficiently as we can to get the required amount of power. You're probably familiar with DC-DC buck converters if you've worked with Arduinos or Raspberry Pis for a while – they are those little variable voltage supplies that can drop 12V to 3.3V or 5V efficiently. You may even have noticed that some of those types of board are labelled as 'buck-boost' converters. The word boost in that context means that the supply can boost the output voltage higher than the input – in other words, you can convert 12V or less input into a 21V output.

When it comes to monitoring the BMS, you can use an Arduino with a Hall effect probe-based current sensor to keep an eye on how much power the board is drawing at any given time. High current means the battery is charging, while low current means the charger is idle. You can download the Arduino showing how to monitor the power and indicate the charger status using LEDs from **hsmag.cc/issue68**. LEDs are a cheap and effective tool for creating a charge indicator, although, if you wish, you could include an LCD output that displays more detailed charge information.

SENSING CURRENT

The ACS712 current sensor uses a Hall effect probe to measure current passing through a circuit. The output from the sensor is analogue, and with a 5V power source, the sensor should return 2.5V when there is zero current flowing. The ACS712 is available in three versions: 5A, 10A, and 30A. Each of these versions returns a different voltage level per amp of current. The 5A version returns 0.188mV per amp, the 20A returns 0.1mV, and the 30A returns 0.066mV. To measure the power used by the BMS, the power should be connected in the positive line between the buck-boost power supply and the BMS. It's worth noting that this method type of sensor can pick up a bit of electrical noise, so it's best to use a threshold cut-off to stop the LEDs from showing the wrong information.

Below
Fitting everything inside the case can be tricky. Remember to allow plenty of space for air flow, add a fan if necessary, and insulate any exposed connections. Hot glue can be useful to help with the layout and wire management, but hold boards in place with mechanical fixings like machine screws. If the system overheats, you don't want critical items like boards breaking free of their fixings and floating around free inside the case



HackSpace magazine meets...

Tim Hunkin

The not-so-secret life of automata



Think of automata, and the image that pops into your head will most likely depend on your age. If you're a child of the 1980s, you're most likely imagining the seaside amusement from the film *Big* that turns a child into Tom Hanks. If you're a child of the 1880s, you might be thinking of the Mechanical Turk, the machine that terrified audiences by seeming to play chess against a human opponent. Others still may think of the works of Tim Hunkin. He's been building machines to inspire and delight for decades, and now has two arcades full of his own creations: Novelty Automation in London, and the Under The Pier Show in Southwold. We caught up with him to chat about precision, engineering, and the joy of the mechanical in a digital world.


SENTINEL



Right
If you ever want to learn about cutting metal, Tim's the man to teach you



HS Your essay on precision reminded me of something my father used to say whenever I was helping him with DIY jobs: "We're not building pianos."

TH I like this one: "Jack of all trades, master of none – but oftentimes better than master of one."

HS Is there still room for the Jack of all trades? It feels like graduate jobs have replaced apprenticeships as entry-level jobs so, by default, any new engineer, programmer, or whatever will have three or four years' specialisation behind them already.

TH Good question. You're right that big companies now only hire people who tick the boxes. But it's much less proscribed for tiny businesses – which is now where most innovation comes from. Will Jackson, a friend who runs Engineered Arts, a successful robot company, generally only employs school leavers who pass his tests to show their aptitude. For something like robotics, it's essential to be able to move seamlessly from hardware to software when solving problems. So, although it often feels like being an old hippy or being a Jack of all trades, I still feel it's very valuable – if, for nothing else, to reduce development costs.

HS You say in your essay on precision that, "Sadly most engineers have to be intensely conservative about their designs. It's expensive to try anything new, and even more expensive to rectify something that doesn't work. Also, because they rarely make their own parts, any detail that's imperfect is alarming to them, even though the part might still work perfectly well."

On this logic, I'd argue that most people who make things for their own consumption – makers, home tinkerers – would be inventors, rather than engineers. Do you agree? And do you see yourself as conservative in your own designs?



Above
Welding is just one of many maker skills that looks easy when Tim does it

TH I hadn't thought of it like that, but I guess it's basically true. It's partly because most engineers work on big projects where the huge scale needs mathematical calculations and any mistakes are hugely expensive. Generally, people think of engineers as boring – I don't, but it can be useful.

> Generally, people think of engineers as boring - I don't, but it can be useful

Trapped talking to someone tedious at a party, saying that I'm an engineer is a sure way to get them to move on.

I often try out new stuff but am also quite conservative. I think the reason why there aren't any other 'homemade' arcades is that it's so hard to make the machines reliable enough. Many of mine have now been used over 100,000 times. So, personally, I am both – inventor and engineer.

HS I was amazed to discover on a recent visit to the Science and Industry Museum in Manchester that Joseph Whitworth built a system that could →



Above
"In the past, all humans used tools to build things, but now they can't even build their own nests"

measure to an accuracy of one-millionth of an inch in 1840. How does the average garage-based inventor/maker get anywhere near this level of accuracy?

TH I don't – I rarely work to more than .02 mm (the resolution of my lathe and mill DROs). At university, in 1970, I was taught about tolerances and told not to specify more precision than necessary because it added cost. With today's CNC tools, this is less important and designers often lack the practical experience to know what tolerance is needed for a part, so specify tight tolerances just to be safe.

HS "Builders use laser levels to make walls and floors accurately vertical and horizontal because it saves time – there's less measuring to do. This is partly why it's often cheaper to knock down an old house and build a new one, rather than to restore the old one."
This is something I'd never thought about before. We've obviously gained a lot through increased accuracy, but do you think we've lost anything in the process?

TH Yes. Vernacular architecture has a charm no architect can match. Many more tourists flock to 'picturesque' villages than to 'modern' architecture – and none to housing estates, today's closest equivalent to a village. In my 20s, I made some furniture and often tried to cut everything with a jig-saw (avoiding using a table or bandsaw) just so, when you run your finger along an edge, it feels uneven and handmade.

> Once someone has a body of work, no one asks to see their qualifications

HS How did someone with no apprenticeship or degree end up being trusted to design gates to separate tigers from delicious, meat-based members of the public?

TH I'd made an elaborate clock for the zoo a few years before, and became friends with the architect. Designing the tiger enclosure, the architect found the only 'off the peg' vertical gates were German electrically-operated ones that cost a fortune. So, he asked me if I could come up with a simpler idea. So, really, the answer to your question is that once someone has a body of work, no one asks to see their qualifications. I do actually have an engineering degree, but no one has asked to see it.

HS Is that something you'd offer as advice to home- and shed-based makers? And what's the best way to build up that body of work, and make sure it attracts the right eyeballs?

TH It wouldn't work for most people. But it can for anyone who's driven to pursue stuff that interests them, rather than to make the most money. For them, my advice is simply don't give up – it always takes longer than you think.

HS Many of our readers will know you already from *The Secret Life of Machines* [at this point we'd direct anyone who hasn't done so already to go





Above
The Secret Life of Machines **explores the technology that's all around us**

Left
Each episode of ***The Secret Life of Components*** **dives into things you've probably never thought of. Imagine what life was like for makers before we had standardised screw threads, for example?**

Right
Prototyping is perhaps the area of making we enjoy the most





to YouTube and watch every single episode of Tim's series, *The Secret Life Of Machines*, swiftly followed by *The Secret Life of Components* – for most people, they're better than an engineering degree]. Would it be possible to make *The Secret Life of Machines* today, or are modern devices too boringly dependent on electronics?

TH I'm often asked, or encouraged, to do an update with today's machines. The problems with today's machines are that there's less to point a camera at, and that hacking into the software would be an essential ingredient, but I don't have the skills or interest to do it.

HS There's something Hogarthian about a lot of your automata, in the way it pokes fun at the world around us. Do you see yourself as a mechanical satirist?

TH I'm quite happy to be described as that, and I am also a huge Hogarth fan. But there is another strand that's important to me, which is the importance of working with your hands. There are far too many people who spend their lives preaching about their pet subjects, so I prefer just to demonstrate what's possible.

HS If I'm reading your website right, you made your first automaton in 1980, for Cabaret Mechanical Theatre in Covent Garden. Did they approach you, or the other way round? What did that first automaton do, mechanically?

TH Actually, I started making automatons when I was a kid. I think I made 'Gladys the burglar catcher' when I was 14. As a teenager, I loved London's many surplus shops and made bits of 'kinetic art' thinking they were more sophisticated than my earlier machines. But when I moved to the country, and got my workshop in my late 20s, I returned to making automata, all out of wood at first. I used to take them to the local fairs and steam rallies. But they took up a lot of space so, when I heard Cabaret was moving from Falmouth to London, I got in touch.

HS How do people react when they see Novelty Automation? Loads of people spend their time staring at screens, I'd imagine that even though the technology is old, many of your visitors won't have seen anything like it before.

TH At first, most people were baffled by Novelty Automation and we lost money. Fortunately, after a year or two, Atlas Obscura got interested, which led to more videos and publicity. Now people know about the place, it's a great advantage being different from anything else. It's also fortunate that the short videos visitors take of each other using the machines are perfect for TikTok.

> "The problems with today's machines are that there's less to point a camera at"

HS Do you have a favourite *Secret Life*, whether Machines or Components? If so, why?

TH I do have a favourite arcade machine at the moment, 'Trust Wildlife?' I spent over a year making it, and had no idea what it would be like until I added the sound track in the last month or two. I used my own voice, greatly improved with Adobe Audition, which was fun. But really, why I like it is that, a few months after putting it on the pier, I was showing a couple of visitors around and, while they were inside, I realised something that had never occurred to me before – the seagulls say pretty much exactly what I think of the world.

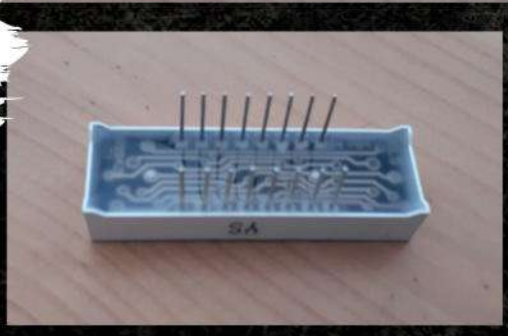
FEATURE

IN THE WORKSHOP: Four-digit display

By Andrew Gregory

Working with a cheap (in terms of money – not in terms of time) display





Above
The letters YS are the only identifying marks on this display

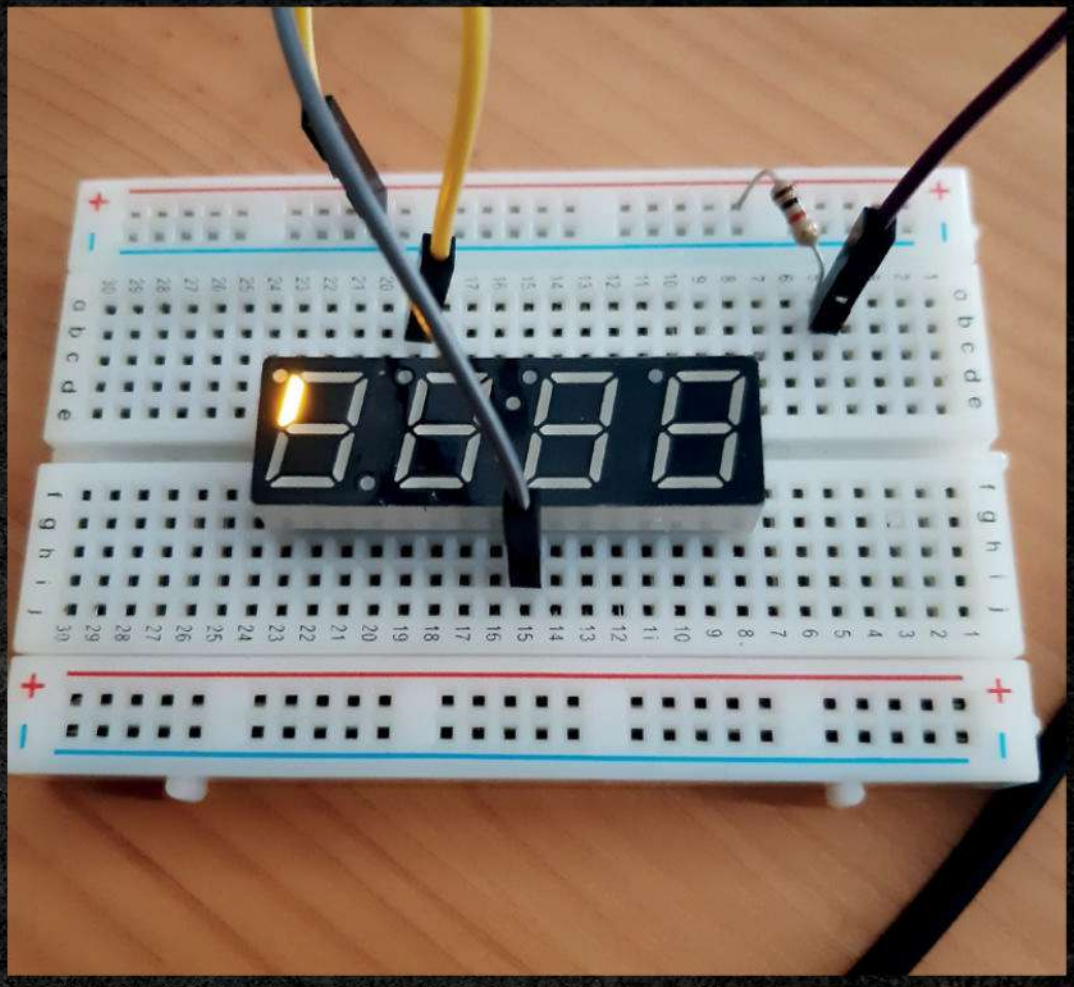
Right
No numbers on any of these pins. We're just guessing here

A while back, I bought a four-digit, seven-segment display, hoping to put it into a clock, or perhaps a simple counting device. I'd expected to be able to connect it with power, ground, and two data wires, but this display has 16 pins, none of them labelled. I would not be able to simply plug this into a Raspberry Pi Pico and throw MicroPython at it.

Before I even get close to programming with this, I'll need to wire it up. There are plenty of tutorial examples on the internet of wiring a four-digit, seven-segment display with dots between each digit (8.8.8.8.), but I couldn't find any for the version I have, with two extra characters and two extra pins (8.8.:8.'8.).

Some electrical components are unidirectional, such as diodes (including LEDs) and electrolytic capacitors. These components will have one leg longer than the other to denote which way round it's supposed to go, or will have a handy + and - printed on them. This display has none of that. I'm going to have to find which pin is which through trial and error.

Now, to the bench power supply. The part's datasheet mentions an optimal current of 20 mA; I'll halve that at first and increase it later if I need to. As for voltage, if this is going to be any use at all to me, it's going to have to run off 3 V power from a Raspberry Pi Pico. So, with 10 mA and 3.3 V dialled



Left
Move the ground wire along a pin and the illuminated LED changes. Through trial and error, we've worked out that the yellow wire, which is carrying 3.3V, must be attached to pin 2 in the pinout

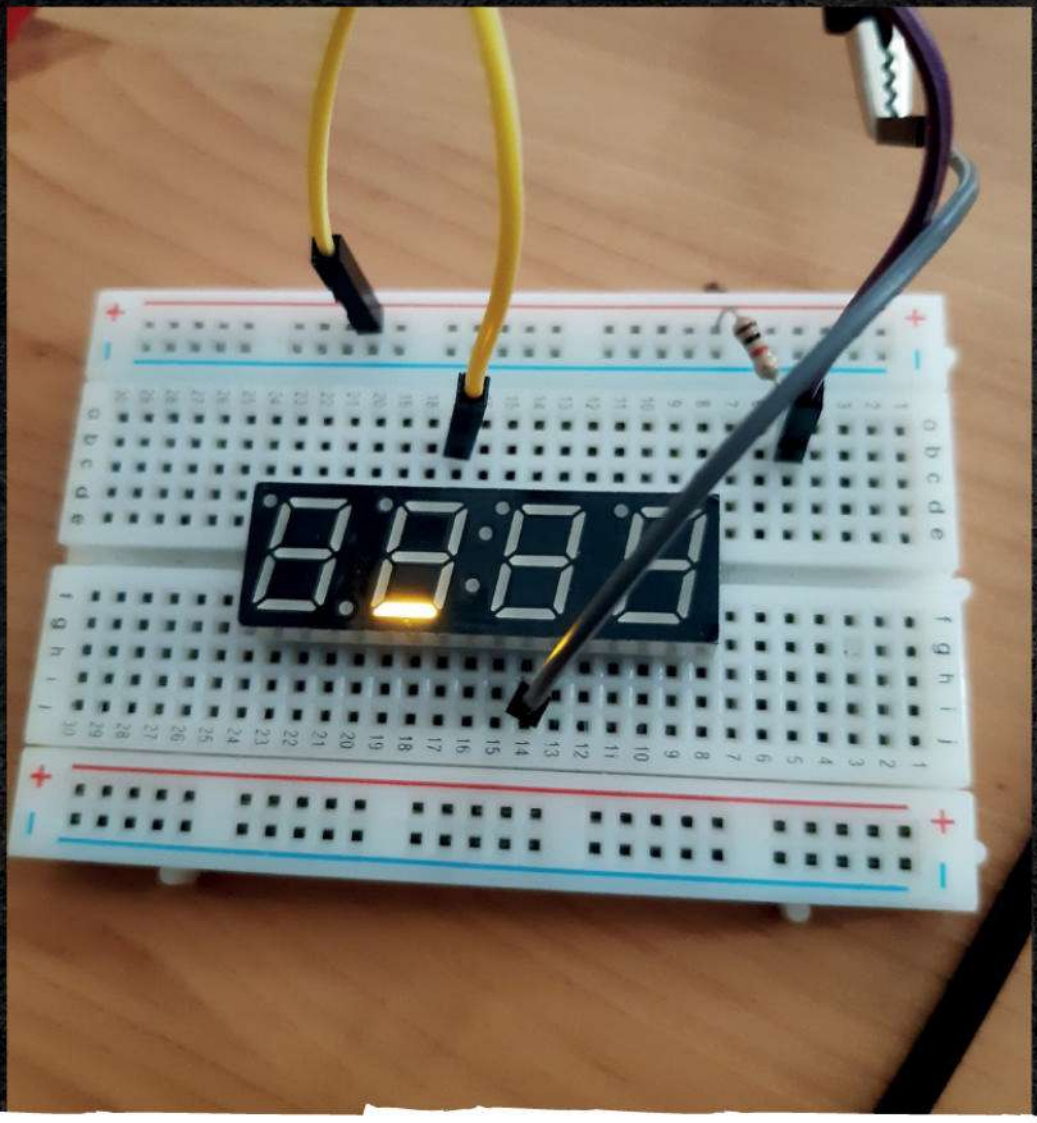
into the power supply, and (to be doubly safe) a 1K resistor to ward off the magic smoke, I started the process of trial and error.

With the bottom-left pin connected to ground, I ran power through each of the eight pins along the top of the display one at a time, repeating the process with the bottom second-left pin connected to ground, all the way through to the bottom-right pin connected to ground. None of them worked.

Undeterred (well, slightly deterred), I remembered that LEDs only work in one direction, so I turned the display around on the breadboard and tried again. Success! With 3V going into the now top-left pin, attaching ground to the third from left on the bottom pin lights up a single LED. Move ground along two pins and a different LED in the same digit lights up; move ground along one more time, and a different LED lights up.

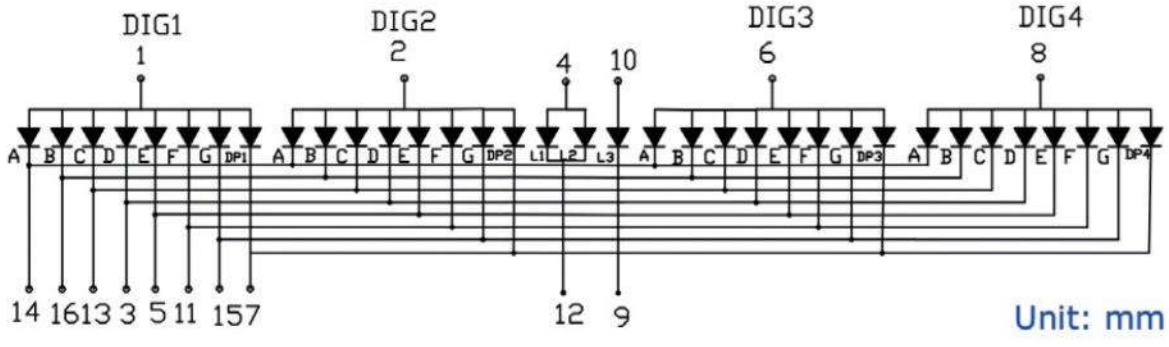
I ran power through each of the eight pins

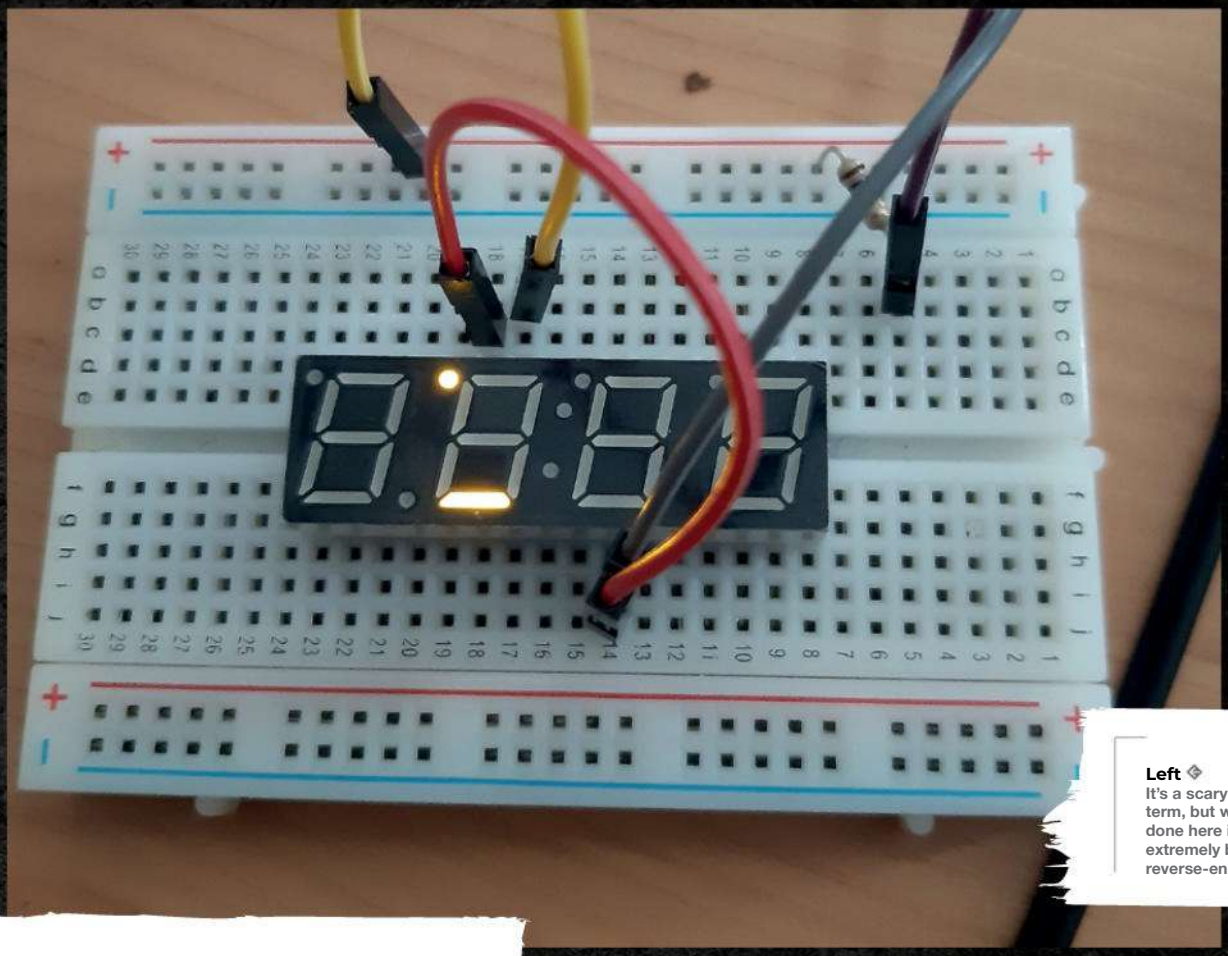
I moved the 3V wire along one space, and tried again moving the ground. No LEDs lit up. But something exciting happened when I moved the 3V wire along one more hole: moving the ground wire along the pins now lit up segments in a different digit. With this success, I added another wire, this time connecting ground to a different pin on the top row of the display. With this, a decimal point lit up. →



Right
Move the ground wire along two pins and the affected digit changes

Below
DIG1–4 probably refers to the four digits of the display. For anything else, we need to dig deeper





Left
It's a scary-sounding term, but what I've done here is some extremely basic reverse-engineering

What this muddled approach tells us is that some pins are used more than once to display a single character. This makes sense, because the display only has 16 pins, while there are 34 LEDs in the display.

Armed with this knowledge, we can make more sense of the pinout on the datasheet. The pinout diagram has annotations 1 to 16 (though not in order, which is what foxed me at first); these will correspond to the 26 pins on the display. By changing the position of the pin that gets supplied with 3 V, we can change which digit contains the LEDs that get lit up; referring to the pinout diagram, we can make an educated guess that we've just identified pins 1 and 2 on the display (we've also identified that pins 1 and 2 aren't next to each other). Or, at least, I think that's what's happened: it's hard when nothing is in English, nothing is labelled, and numbers aren't in numerical order.

By the way, if you search for this yourself, you'll probably find a load of tutorials for using a similar display that only has four pins; this is the TM1637 module, which uses I2C to abstract the wiring to a single connector and the programming to just one Python module. For future reference, this is the smart choice.

Some pins are used more than once

LESSONS LEARNED

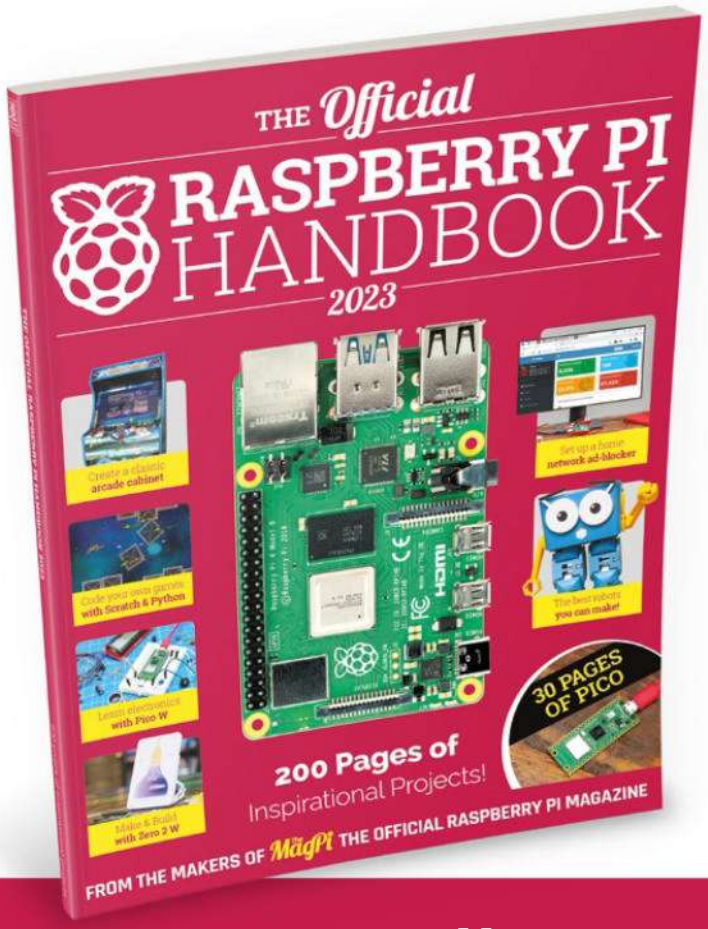
The moral of this story is that you should always check before you buy that the documentation you need is written in a language that you can easily understand.

The other moral is that all this can be avoided if you just spend a few more pennies and buy a module that elides the complicated bits for you. You're not a huge manufacturing conglomerate for which a saving of 20p per component will add up to millions in shareholder value; you're making one, or two, of half a dozen things in your precious spare time and, if spending a tiny bit more money can save you hours of fiddling, it's well worth it. ◘

THE Official RASPBERRY PI HANDBOOK 2023

200 PAGES OF RASPBERRY PI

- QuickStart guide to setting up your Raspberry Pi computer
- Updated with Raspberry Pi Pico and all the latest kit
- The very best projects built by your Raspberry Pi community
- Discover incredible kit and tutorials for your projects







Buy online: magpi.cc/store

HackSpace
TECHNOLOGY IN YOUR HANDS

FORGE

HACK | MAKE | BUILD | CREATE

Improve your skills, learn something new, or just have fun tinkering – we hope you enjoy these hand-picked projects

PG 58
RECYCLE PLA
Use your old 3D prints in new builds

PG 64
GETTING STARTED WITH MICROCONTROLLERS
Take your first steps with programmable electronics

PG 68

POLYCARBONATE
A challenging filament for challenging environments

PG 70

BLUETOOTH KEYBOARD
Customise your computer with a Raspberry Pi Pico W

PG 52
SCHOOL OF MAKING
Start your journey to craftsmanship with these essential skills

52 KiCad

PG 76

PLANETARY GEARS
Compact drive chains that are easy to 3D-print

PG 80
MACRAMÉ
How do you make things with string? We can knot

KiCad libraries, symbols, and footprints

In this third part of a series of articles around PCB creation with KiCad, let's look at adding or importing custom libraries, schematic symbols, and component footprints



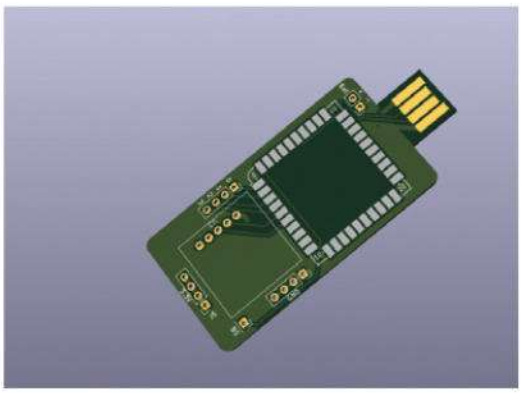
Jo Hinchliffe

@concreted0g

Jo Hinchliffe is a constant tinkerer and is passionate about all things DIY space. He loves designing and scratch-building both model and high-power rockets, and releases the designs and components as open-source. He also has a shed full of lathes and milling machines and CNC kit!

In the first two parts of this KiCad series, we've covered enough to make a basic PCB suitable for simple circuits, or maybe a breakout board. In this part, we are going to look at some next steps that open up the capabilities of what you can make in KiCad. We'll explore both creating libraries and contents from scratch, but also importing and using component footprints and schematic symbols from other sources. We'll also improve the quality of the boards by using flooded areas for common connections, such as all the circuit points that are connected to ground.

We've used a relatively simple design to show these techniques – making a PCB that essentially has two modules on board. Quite often, when making, we work with electronics modules on breadboards and a custom PCB can be the perfect way to take a breadboard project to a permanent home that's more rugged and usable. We aren't going to go step by step through the making of this board, but the project files are available at **hsmag.cc/issue68**, and the knowledge and techniques we used in the first two parts of this series, in combination with this section, should give you enough knowledge to recreate this project.



Right
Our completed board, ready to receive a Solder Party Stamp and a Pimoroni BMP280 module

We've chosen to use a Solder Party Stamp and a Pimoroni BMP280 module connected together so that the resulting board can be used to measure and log temperature and barometric pressure. The Solder Party Stamp is an excellent board that has an RP2040 at its heart, and is operationally similar to a Raspberry Pi Pico. The RP2040 is fully broken out to header pins which are castellated, so you can also solder the Stamp onto a recipient PCB's pads without having to use header pins. The Stamp also has the USB connection broken out as well as on-board LiPo charging. This means if we add a USB connection, we can also add a LiPo cell and make the project stand-alone. The Solder Party Stamp is really well-documented and is open-source. Solder Party has also published KiCad schematic symbols and a PCB footprint for the Stamp, therefore we can use it to learn how to add libraries and import these useful items into KiCad.

To begin, go to the following link where you will find the Solder Party Stamp library components: **hsmag.cc/StampFootprints**. Click the drop-down menu on the green Code button and then select the Download ZIP option to download the libraries. Unzip the files somewhere on your machine. In the collection of folders you just unzipped, cut and paste the entire **KiCad** folder (N.B. not the **KiCad 5** folder) to wherever you want to store your additional

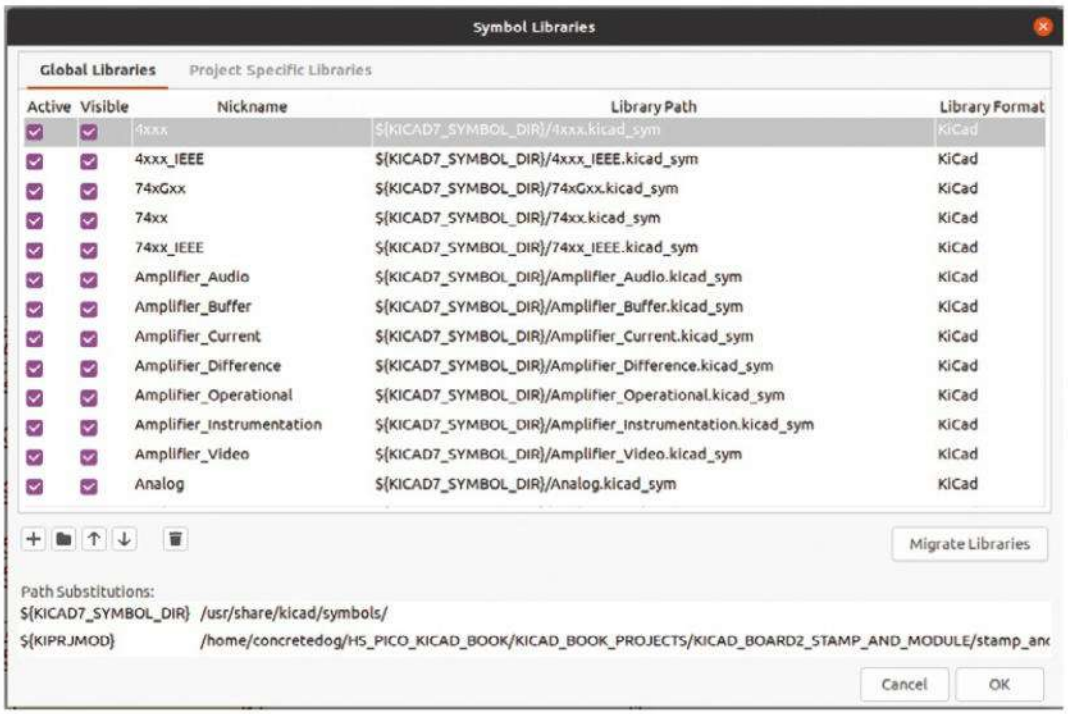


Figure 1
The Symbol Libraries dialog where we can add or remove schematic symbol libraries

QUICK TIP

In the project schematic, we have sometimes used net labels to create connections between symbols rather than direct drawn wires. We'll cover this in some detail in an upcoming article.

external KiCad libraries. We have a folder set up in our home directory for this.

Open KiCad 7 and, in the main page, click the Preferences drop-down menu and then select the Manage Symbol Libraries option. This should open a window with two tabs: the Global Libraries tab and the Project Specific tab, (**Figure 1**). Ensuring you are on the Global Libraries tab, find and click the small folder icon. Navigate to the folder we downloaded, extracted, and copy-pasted and open it to find a folder called **KiCad_stamp_lib**. Open this folder and select **RP2040_Stamp.kicad.sym** and then click Open. You should now see a new library listed at the bottom of the Global Libraries tab called RP2040_Stamp. If you create a new project and open the Schematic Editor, you can now use the 'Add a Symbol' tool to place a Solder Party Stamp symbol into the schematic. You can do this by searching for RP2040 and making sure you select the symbol from the RP2040_Stamp library (rather than the stock RP2040 symbol) or by scrolling down the list of libraries, selecting RP2040_Stamp, and then selecting the RP2040_Stamp symbol.

It's a similar experience to add a footprint library. Again in the KiCad landing page, click Preferences and then Manage Footprint Libraries. Again, on the Global Libraries tab, click the small folder icon (**Figure 2**). Navigate to the folder we downloaded and find **KiCad_stamp_lib** – open that folder once more but, this time, select the **RP2040_Stamp.pretty** folder and click Open. You should see three files inside, but you don't need to select any particular one – just click Open again. Now, back on the Global Libraries tab, you should be able to scroll down and see an RP2040_Stamp library entry. You can check that this has all worked by associating the correct stamp footprint to the RP2040_Stamp symbol we placed in the Schematic Editor, and then you can open up and import the part into the PCB Editor. If you need a reminder on how to do those tasks, check out the first and second part of this series in HackSpace #67 (**hsmag.cc/issue67**).

> **If we add a USB connection, we can also** add a LiPo cell and make the project stand-alone

Using the BMP280 module gives us an opportunity to learn how to make both a custom schematic symbol and a custom footprint for the →

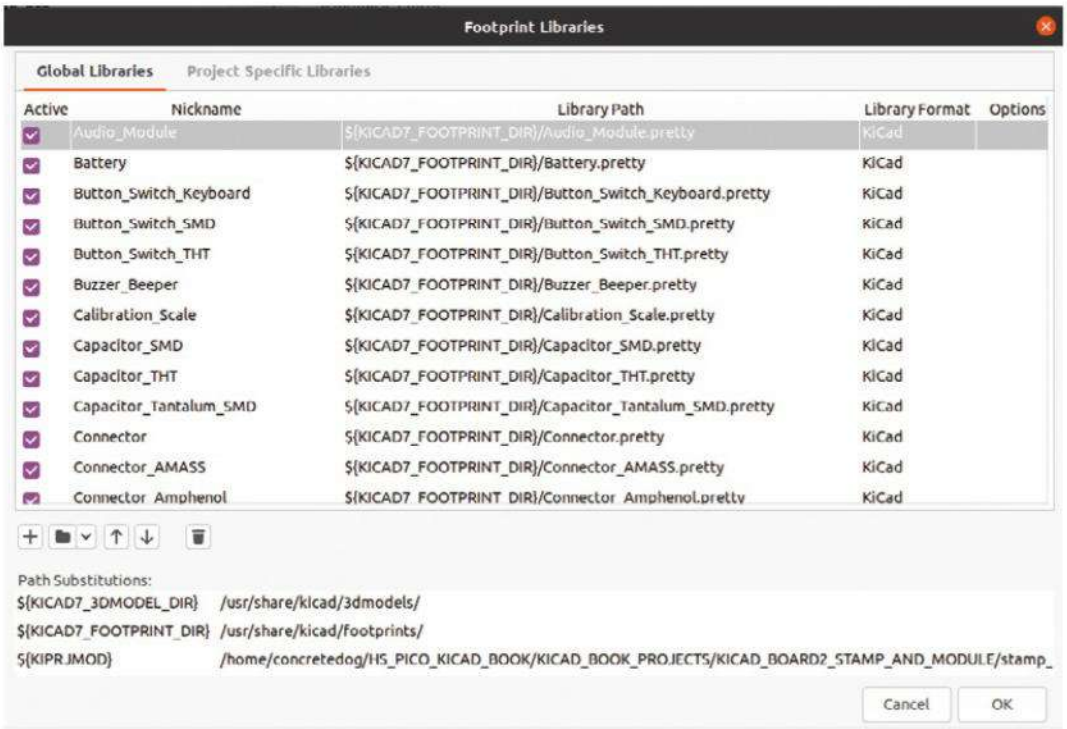


Figure 2 The Footprint Libraries dialog where we can add or remove PCB footprint module libraries

module to use in our board. To do this we will create our own custom libraries to contain these parts, and others in the future. Let's begin with a custom schematic symbol. In the Schematic Editor, click the 'Create, delete and edit symbols' tool button. This opens the schematic Symbol Editor.

In this new window, click File and then select New Library from the drop-down menu. You should now see a small dialog box called Add To Library Table. In this box, you can select to add your new library to the Global table. This means any project in KiCad can access this library; alternatively, selecting Project means only this KiCad project can access that library. As the BMP280 is something we might use in other projects, let's make sure 'Global' is highlighted and then click OK. You'll now be asked to give your library a name – it can be anything you want, so name it, making sure to leave the '.kicad_sym' part of the file name intact, and click Save. You should now see your new symbol library name highlighted on the left-hand side of the Symbol Editor window. This means that this is the active library, so that when we select to create a new symbol, it will be stored in this library automatically. Click File and select New Symbol… from the drop-down menu. You should see a New Symbol dialog appear. Give your symbol a name that will be used in the library list – make it a useful name that reflects the part. We went for 'pi_bmp280'. Clicking OK, you should now see that a 'U' has appeared in the Symbol Editor window and that the name of the symbol now appears in the active library in the list on the left-hand side of the screen. The name will have a '*' next to it, indicating that the symbol has not been saved. In the Symbol Editor there are some familiar controls, such as the **F1**/**F2** to zoom in and out. Zoom out a little to give yourself some room and let's get started by adding some pins.

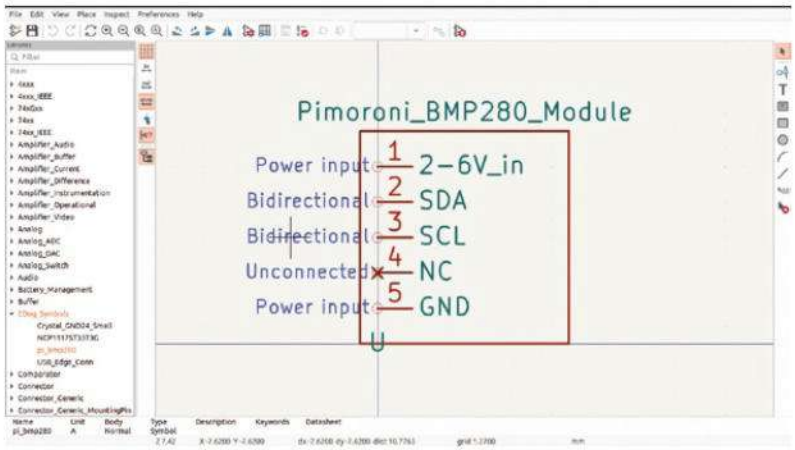


Figure 3 The Symbol Editor window can be used to create or edit schematic symbols

Click the 'Add a pin' tool icon. In the dialog, we can name the pin, give the pin a number, and set the 'Electrical type' and change other settings if needed. For our first pin, let's name it '2-6V_in', assign it pin number '1', and set the electrical type to 'Power input'. Continue to add pins 2, 3, 4, and 5, labelling them as you can see in **Figure 3**. As you place pins, notice that you can use the generic hot keys **M** for

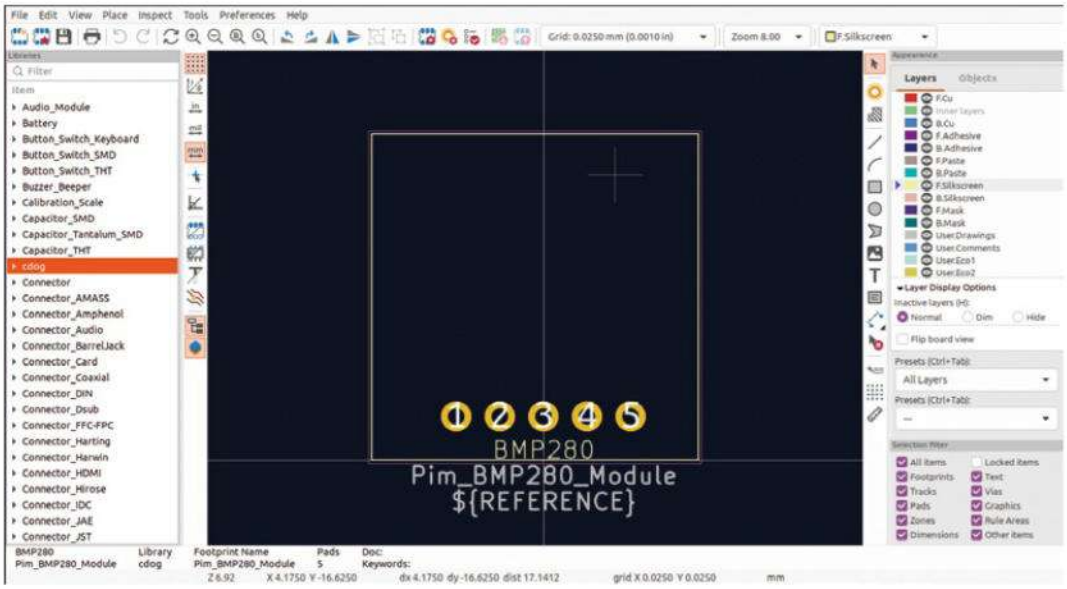


Figure 4
The Footprint Editor window can be used to create or edit component footprints

move and **R** for rotate, similar to the Schematic or PCB Editors. Once you have all your pins created, you can add a text label using the 'Add a text item' tool. This is useful so that you can identify the symbol quickly when looking at a schematic. Finally, let's draw a bounding box around our schematic symbol so that everything is neatly grouped together. Click the 'Add a rectangle' tool, and draw a rectangle over your design. Click the Save icon

> Give your symbol a name that will be used in the library list **– make it a useful name that reflects the part**

in the top-left corner of the screen, and then close the Symbol Editor window. You can now go into the Schematic Editor and use the 'Add a symbol' tool to find and add your first custom symbol.

Next, we need to make a new footprint library and footprint for the Pimoroni BMP280 module. To begin, we first need to open the Footprint Editor (**Figure 4**). This is available either from the main KiCad project window, or indeed can be launched from the 'Create, delete and edit footprints' tool icon in the PCB Editor. Similar to the Symbol Editor, once you have the Footprint Editor open, the first thing to



Figure 5
Editing the pad size and shape using the Pad Properties menu

do is to click File and then select New Library from the drop-down menu. Again, you can select to add a new library to the Global or Project table – select Global and name your library. Once the new library appears in the list, highlight it and then click File and select New Footprint. A dialog appears and we can name our new footprint. We called ours 'Pim_BMP280_Module'.

We also need to select from the drop-down menu whether this is an 'SMD' or 'Through Hole' component. Selecting 'Through Hole', we are now ready to add the pads and other parts of our footprint. Similar to the PCB Editor, we can set the grid resolution and, as the pins on the BMP280 module are spaced at a standard 2.54 mm pitch, it is worth setting the grid to this initially to allow us to easily place the pins. Next, click the 'Add a pad' tool icon. We want to end up with five pads →

QUICK TIP

Note that if you use a schematic symbol or a footprint module from an external library, once it is saved in your KiCad project file, the symbol/footprint is stored in that project. If you moved to another machine with KiCad that didn't have the custom libraries added, you would be able to open the project as usual.

QUICK TIP

If you aren't in the Schematic Editor, you can open the Symbol Editor from the main KiCad project window.

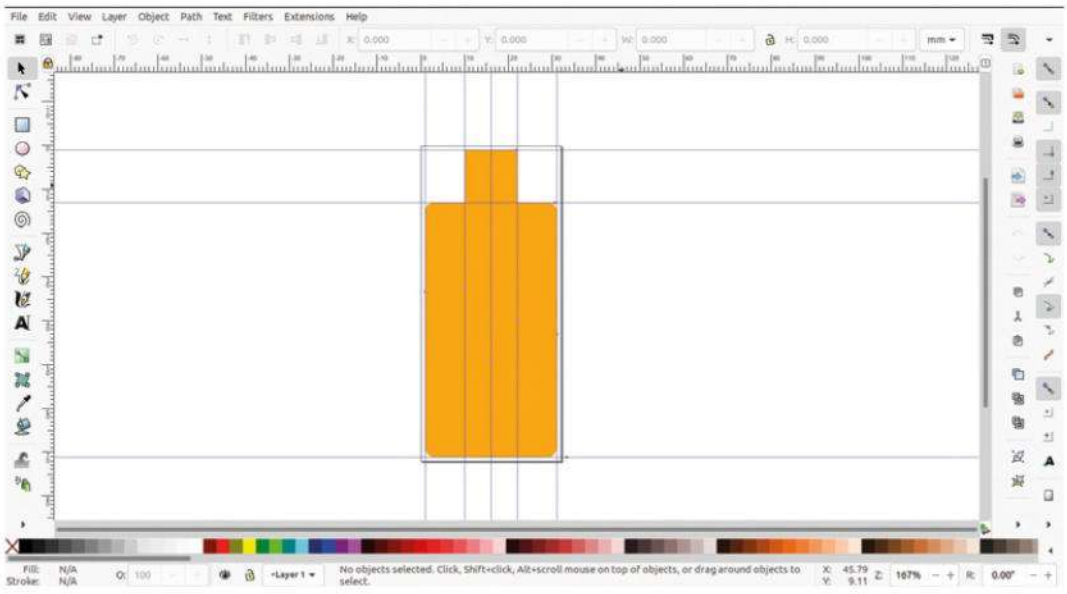


Figure 6 Using Inkscape to create an accurate graphic for the PCB edge-cut geometries

> **Many components will have physical package** dimensions listed in their datasheet

USB ON BOARD

We've now nearly got everything we need to make the Stamp and BMP280 board! You might have noticed in the main image that we have added a USB edge connector directly onto the PCB. This is a cheap and cheerful approach to adding USB without adding any extra components, although we will need to add around 1 mm of material to the back of the 1.6 mm thick PCB in this area to actually make the USB connector fit. The reason we added this is to highlight another way of using libraries and components from other, suitably licensed, KiCad projects. In this example we found a project, the USB Armory, which, in an early Mk1 version, used the USB edge connector on PCB approach. The USB Armory is an open-source project and we can download the project repository here: **hsmag.cc/USBArmory**. Once downloaded, unzip the folder and use KiCad to navigate to the **hardware** folder, then **mark-one**, and then open the file **armory.kicad_pro**. Once open, move to the PCB Editor and select the USB edge connector footprint and press **E**. In the Footprint Properties window, click the Edit Footprint button. This should open the footprint in the Footprint Editor. You might get a warning that the footprint was made with an earlier version of KiCad, but saving the footprint in the editor should clear this warning. You should see that the footprints pads and the silkscreen line (which doubles as a guide for the edge of the PCB board cut out) are now opened in the Footprint Editor. Across the top of the editor you should see a warning that you are currently only editing the footprint within the current project. You can use File then Save As to rename and save this footprint into your custom library that we created earlier. As that library is created on the Global table, this USB edge connector footprint is now available to use in any project. We edited our version a little, labelling the pads more clearly, and saved it to our library. Making a note of each pad's connectivity, we then repeated the earlier approach to create a custom symbol in our library to represent the USB connector in the schematic.

labelled 1 to 5 moving from left to right. The easiest way to do this is to count two grid spacings out from the centre datum line and then click to place pad 1. Notice that the 'Add a pad' tool then increments the pad number so the next previewed pad is labelled '2'. Move one pad to the right of the pad you just made and click again. Continue to do this until you have a neat row of five pads.

Reverting to the general 'Select items' tool, hover over pad 1 and press the **E** key to open the 'Pad Properties' dialog (**Figure 5**). In this window you can change the geometry of the pad, the size of the hole, and many more options. We've found that increasing the pad size slightly from the default, and increasing the hole size, works well for soldering header pins between modules. You may have your own preferences, but we edited each pad to be a 1.8 mm circle with a 1 mm hole. With the pads created, we next need to add a silkscreen item that represents the physical area the module will occupy. Many components will have physical package dimensions listed in their datasheet, but that's not always the case with modules that are really designed for prototyping and breadboard use. In these cases, some investigation with a pair of callipers is a good approach to get some dimensions of the package. Measuring the Pimoroni BMP280, we realised a 19 mm square area with the pin pads 2.54 mm from the edge gave us a slightly oversized and therefore safe margin for the module. Select the 'F.Silkscreen' layer on the right-hand side of the screen and then use the 'Draw a rectangle' tool to draw this and position it correctly. Finally, let's add

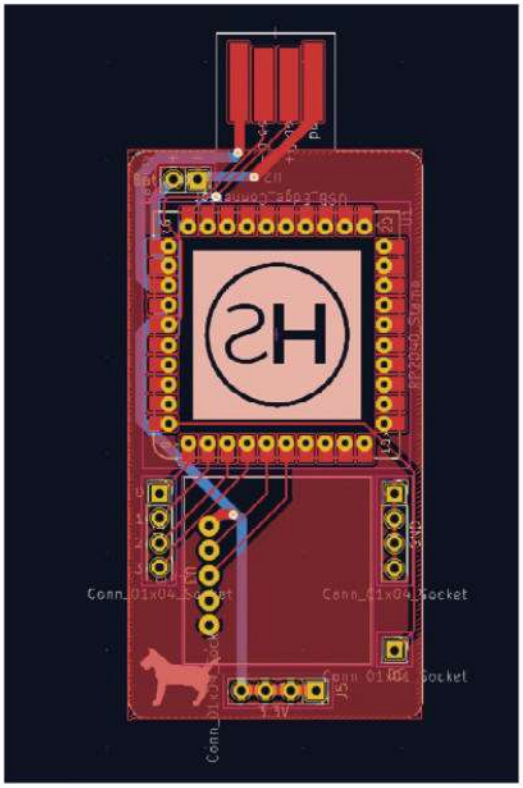


Figure 7
Flooding the board to connect all the GND connected pads

another square on the 'F.Courtyard' layer that sits just slightly outside the silkscreen layer square we just created. You can achieve this by setting the grid to a very small spacing value such as 0.01 mm and then drawing a rectangle away from the silkscreen rectangle to avoid it snapping. Make the new square 19.4 mm and then you can use the **M** key to move the square into position. This new square in the front courtyard layer provides a service called the 'DRC' or 'Design Rules Checker' with a boundary that shouldn't be overlapped. This means that later in the process, if a component is overlapping, this boundary on the PCB when we run the DRC will be highlighted as an issue. We'll look at using the DRC in the next part of this series.

To create the board outline, we decided to not use the graphical tools in the PCB Editor but, rather, we imported an outline we drew in the free and open-source Inkscape application (**Figure 6**). Whilst the included KiCad tools are excellent, Inkscape can offer some advantages when designing graphic components. We drew a simple outline object for the board in Inkscape, saved it as an SVG file, and then imported it into KiCad. To do this, click File and then select Import > Graphics… from the drop-down menus. In the Import Vector Graphics File dialog, you can navigate to the file and select the working layer to import to. Setting the graphic layer as 'Edge.Cuts', notice that we can also set an Import Scale value. In this instance we designed the board outline to be the correct size in Inkscape, so we leave the import scale at 1.00 so that it imports at its original size. This function is useful. However, if we have an oversized graphic or logo to import to a silkscreen layer, like the HackSpace logo, we've reversed in Inkscape and then imported it onto the back silkscreen layer. We'll look at creating PCB artwork and components in Inkscape in more depth later in this series.

Finally, another technique that we've used on this board is to create flooded copper zones attached to a net label. This is an excellent way of automatically connecting groups of common connections (see **Figure 7**). You can create complex systems with lots of different flooded areas with differing connectivity, but in this project we have just used one flood on the front copper surface that connects all pads attached to ground. To do this, once the board is laid out, we used the 'Add a filled zone' tool. When you select this tool, you then left-click to start to draw a fill area over your board design, ensuring you are on the correct layer. A dialog appears and you can select the net connection from the list, so we selected 'GND' (**Figure 8**). Leaving all the other settings at the default values, we drew a rectangle over our board. Don't worry too much about accuracy as the flood will only appear inside the edge-cut geometry. Drawing three points of your rectangle you can then right-click and select 'Close outline' from the drop-down menu. The rectangle (or other shape you drew) should now flood, and you should see that all the previously disconnected GND pads are now connected together – neat!

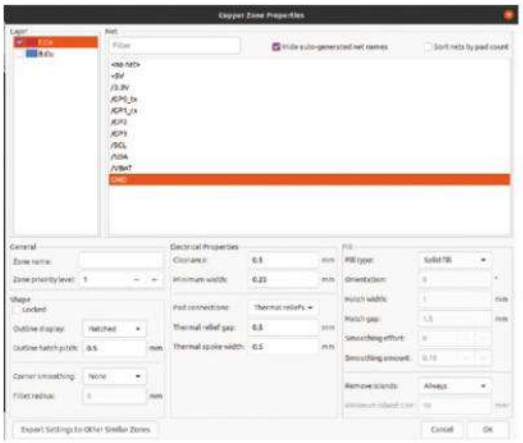


Figure 8
Selecting the net from a list to which the flooded copper zone will connect

QUICK TIP

Note that the schematic symbol pin numbers directly link to the assigned footprint pad numbers, so you must make custom symbols and footprints match.

Recycling PLA

Turn filament scraps into beautiful new objects



Ben Everard

@ben_everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

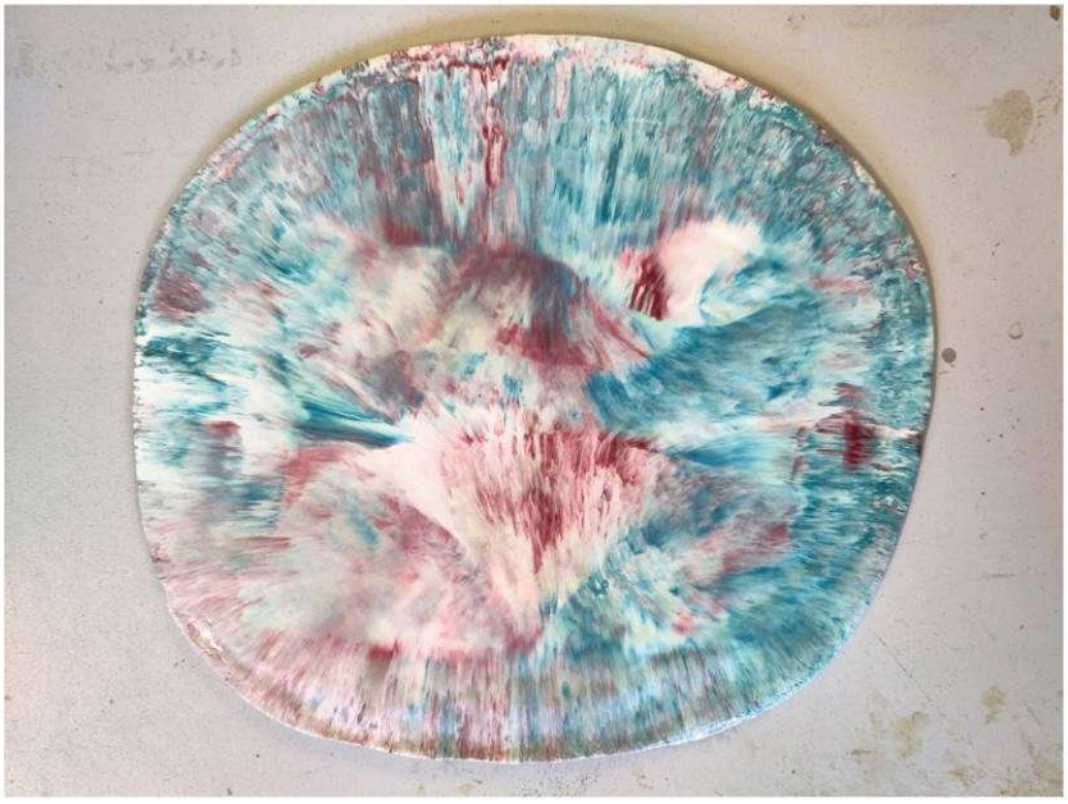


Above & Right In some early tests, we ended up with very blocky colours, but we fixed this by remelting the sheet and using the fold-and-twist method

Let's be honest, if you have a 3D printer, you probably generate quite a bit of plastic waste. There are ways to minimise this – and you absolutely should – but you can't eliminate it entirely. There will always be some failed prints, and sometimes the first version of something you've designed isn't perfect (and neither is the second, third, or fourth). Can we reuse this waste plastic rather than throw it away?

This project started because of an off-hand comment in our local hackspace: "Wouldn't it be nice if you could feed failed 3D prints back into the printer." This is something that comes up fairly regularly in the 3D printing community, and there are a few projects that attempt to do this, either by reforming scrap PLA into filament, or by making an extruder that can be fed with shards of plastic rather than a spool of filament.

There are a couple of key problems you have to overcome. The first is that you need quite a bit of equipment. At the very least, a shredder to break up the plastic, and an extruder to melt this plastic and squeeze it through a nozzle (either on the printer itself or to make new filament). Both of these are reasonably costly machines. Once you've got the stuff, it's a fairly time-consuming process. And finally, if you get any particles of metal or anything that's not PLA in the mix that are larger than 0.4 mm, then you'll jam your nozzle when you try to print it.



Overall, it's not impossible, but it's not entirely trivial, and the amount of effort is quite large to create a spool of filament that you could buy for under £20.

The project almost died there until we realised that the 3D printer isn't the only machine that we used plastic in. We also used acrylic sheets in the laser cutter. What if, instead of making PLA filament for the 3D printer, we made PLA sheets for the laser cutter?

We've got a heat press (for pressing vinyl and sublimation prints onto fabric), which should be able to squeeze soft plastic into a sheet if we can work out the appropriate settings.

First, though, we needed some scrap PLA. We left a box in our local hackspace asking for donations, and they quickly turned up. Last month we also spent a long time testing the Prusa MK4, and that meant a lot of printing. While we try to test printers with useful prints, there are only so many things we need, so a large number inevitably end up as scrap.

There are a lot of variables to tune, but the first ones were part size and temperature.

Part size affects a few things – mainly aesthetics (more on this later) and how fast it melts. We don't have a shredder, so getting small parts is mainly a case of using pliers, a vice, a bandsaw, and a hammer in various combinations to break up bits into smaller pieces. After much sweating and swearing, we've finally come to the conclusion that you don't actually need to break up the plastic in most cases. The best →

> We've finally come to the conclusion that you don't **actually need to break up the plastic in most cases**

SAFETY

Plastics give off fumes when they melt. Some give off more dangerous fumes than others, and some give off more than others.

Obviously, this includes your 3D printer. However, we're melting a lot more plastic a lot quicker than you do on a 3D printer, so it's going to give off more.

Most of the fumes given off by PLA aren't known to be particularly toxic, but there's a lot unknown about exposure to these sorts of chemicals. The risk is increased significantly if you do this regularly.

There are a few things you can do to increase your safety. Most obviously, you can increase your ventilation – open a window and ideally point a fan at it to increase airflow. You can also work with the coolest possible temperatures. PLA emits a lot more fumes once temperatures get to around 200°C.

Finally, you can wear a respirator. It needs to be both correctly fitting and rated for filtering volatile organic compounds (VOCs) to be effective.

The exact combination of measures you need will depend entirely on your setup, so it's up to you to understand the risks if you choose to follow this method.





Above & Right
We love the combination of black and orange, but it wasn't really working until we tried the swirl pattern

solution is actually to blast it with a hot-air gun until it's soft enough for the initial squish in the heat press.

We found that we got reasonable results between 180°C and 200°C, with the optimal being around 190°C.

To protect the press, we used baking sheets on both sides. Our press only heats from the top, so we had to periodically flip it to ensure it melts from both sides. Once it's been squished to the desired

> Blast it with a hot-air gun until it's soft enough **for the initial squish in the heat press**

thickness, take it out and place it under something heavy to keep it flat while it's cooling (we found that a few hefty books worked well).

The result is a fairly flat sheet that laser-cuts reasonably well.

As a general method, this works, but it has a couple of problems:

1) depending on the parts you started with, you can get a lot of bubbles trapped in the sheet

2) the colours can be very blobby, and it's not the most aesthetically pleasing effect.

The solution to both of these problems is to mix the plastic a bit more. For this, you'll need some heat-proof gloves. We've tried leather work gloves and silicone oven gloves and both work fine. The only ones we've had trouble with are work gloves with some sort of flexible rubber grips on them which melted into the plastic.

To get rid of the bubbles, simply folding and twisting the plastic a few times seems to work well. This might feel like it'd introduce air, but it really doesn't.

If you're just here to create functional parts, then you can finish here. You should have some reasonably consistent, smooth sheets that you can laser-cut. They're also easy to work with using hand tools as well.

Beauty is, as they say, in the eye of the beholder, and what we find interesting might not be what you find interesting. That said, there are some basic techniques that you can use to create an interesting look.

The first and most obvious thing is to think about your colours. If the colours don't work together, then the pattern doesn't really matter, it's not going to look good. You can either hold onto the PLA until you have some more colours, or you can use them to create functional parts or test pieces that don't have to look good.

We found two ways of creating interesting-looking prints.

The first is to have a roughly even mix of two colours to start with. You can then melt these down to a sheet with the colours separated – one colour on one side, the other colour on the other side. You can still mix this up a bit to get the air bubbles out by folding into the middle or perpendicular to the split in the colours. Once everything's fully soft, it's time →

OTHER **PLASTICS?**

We've focussed our efforts on PLA because it's the most popular plastic for 3D printing, but any plastic that can be printed with an FDM printer is a thermoplastic that can be heated and reformed. Should you give this a go with other plastics?

Firstly, you need to think about safety. Different plastics give off different levels of fumes, and these have different levels of toxicity. We would recommend staying away from ABS and nylon unless you are confident in your setup.

PETG might be worth experimenting with if you have enough to try. You will, however, need to work at hotter temperatures.

Obviously 3D prints are not the only source of plastic that can be recycled. We've seen other makers have success with similar techniques using HDPE (high-density polyethylene) – you can recognise this as it has a number 2 inside the recycling sign. However, we've not experimented with it.



Left
A quick blast with a hot-air gun can really make your bowl shine



Above
Before the laser cutter broke, we tested out some jewellery designs

COMPOSTING

In theory, PLA is compostable. However, the brutal truth is that it's almost impossible to compost it. If you try, you may well be making the situation worse. We would strongly recommend that you don't attempt to compost PLA unless you are very familiar with the specifics of both PLA and the system you'll be using.

PLA will break down in very specific conditions (basically, it needs to be held at over 60°C for an extended period of time). Doing this on a home compost heap isn't impossible, but it is hard. And you need to ensure that all of the PLA is held at that temperature. If you fail to fully compost the PLA, your compost will be full of half-decomposed microplastics that you'll be contaminating the environment with.

There are some industrial composters that do meet the requirements of composting PLA, but it's by no means all of them. You can't just assume that you can add PLA to a food waste collection and it'll be dealt with. In fact, you might make the situation much worse – if the composters aren't expecting PLA in the waste, they may reject an entire batch of food waste and send it to landfill to stop their compost from being contaminated with plastic. If you do want to compost your PLA, it's imperative that you speak with the waste company to ensure that they will process it, otherwise, you could find that you not only don't get your PLA composted, but that you stop a lot of other compostable waste from being composted.

What should you do with PLA? Well, it's complicated. Take a look at the safe disposal box for more information.

to mix. Roll it perpendicular to the colours, then fold it over and twist to swirl the colours together. How much twisting takes a bit of practice to get right. Too little and it's still very blocky. Too much and you just mix the colours together and end up with something fairly brown. Get it right and you can generate some really interesting swirly effects. The amount of mixing you want is often dictated by the size of the thing you're making. If your final objects are small, you'll want a tighter, smaller swirl. If they're bigger, you might want a loose swirl. There's no real substitute for experimentation here.

> "Get it right and you can generate some **really interesting swirly effects**"

The second technique requires smaller pieces with a very mixed range of colours. Mix them as evenly as you can before you melt them and only mix a little to break up the bubbles.

Those are the two effects we're particularly pleased with, but that's just our personal style.

At this point, we had a technique for generating some interesting-looking PLA sheet, and had done some initial tests laser-cutting them. However, the laser cutter chiller at Bristol Hackspace had a 'bang and spark' error that took out the power to the building and hasn't worked since. A new one's on order, but we want to do some more testing before talking more about laser-cutting this material.

Fortunately, though, there's more than one way to work with sheet PLA, and we've been experimenting

with slump-moulded bowls. The idea is really simple. Take an upturned bowl, or similarly shaped object. Heat up a sheet of PLA until it's soft. Plonk the PLA on the upturned bowl and it'll sag down to make a bowl shape. This is very similar to a technique that's often used with old vinyl records.

The amount you heat the PLA sheet affects the final result. If it's very soft, it'll instantly drape over the bowl, and you may find some of the pleats fold together. If it's only a little soft, you can get a gentler, more open bowl. There's no right or wrong here, it's just a matter of what you prefer.

This technique works best if you start off with a circular sheet of PLA. You can either cut this out of a different-shaped sheet, or you can make a circular sheet by depositing a roughly circular blob of PLA in your press after twisting and mixing it.

You can use a heat gun to make minor changes to the shape of your bowl as it cools.

The final thing to give your bowl a glossy appearance is 'flame polishing'. This is where you melt the outer layer of PLA and let it smooth together to create a shiny finish. While this is called flame polishing, it's best to do this with a heat gun. Experiment a little to find the temperatures and flow rates that work best for you, as this will vary a bit between heat guns, and you can also get different effects, from a satin finish to a high gloss.

From a fairly innocuous discussion, we've been down a rabbit hole of plastic recycling and ended up with a technique for generating PLA sheet that's easy to work with and can look great. We're going to further test this technique and this material to see what we can do – we'll report back with our findings in future articles.

Below
A pile of PLA scraps waiting to be pressed into a sheet





Below
You can see in the texture of the bottom-left bowl that we didn't get the bubbles out properly

SAFE DISPOSAL

If you can't compost or reuse your PLA, what should you do? The answer really depends on what your local waste system looks like, and it's impossible for us to give hard-and-fast rules. For example, this author has a different waste stream at home and in the office, and the waste is processed slightly differently.

At home, he has domestic plastic waste. PLA isn't often recycled commercially because there isn't a lot of it in the waste stream. Even if it was, it's pretty hard to get 3D prints to be recognised correctly. You'll need to label them, and ideally, test this label with a plastic sorter.

However, the particular plastic recycling plant that is used by this author's waste collection team has two parts to it. Firstly, it recycles any recyclable plastic. Secondly, it incinerates non-recyclable plastic and uses this to generate electricity that both powers the recycling plant and feeds back into the grid.

In this situation, the PLA (which is plant-derived, not oil-derived like most plastics) essentially becomes a biofuel that happened to be a 3D print along the way. That doesn't necessarily make it great for the environment, but at the same time, it doesn't seem too bad.

In the workshop, the general waste goes directly to a different waste-to-power plant, so PLA should go directly into this, and again, it will be used as a biofuel.

This, however, is the author's specific setup. Your waste streams will be different, and you'll need to familiarise yourself with where different bits of waste end up to know where the best place to put your PLA is.

TUTORIAL

Part 02

Beginning electronics: Introduction to microcontrollers



MAKER

Stewart Watkiss

Also known as Penguin Tutor. Maker and YouTuber that loves all things Raspberry Pi and Pico. Author of Learn Electronics with Raspberry Pi.

penguintutor.com

twitter.com/stewartwatkiss

Learn about microcontrollers using Raspberry Pi Pico. Build a motion sensor which turns on LEDs whenever someone enters a room

In this project, you will learn how to detect when someone enters a room using a PIR (passive infrared) motion sensor. You will learn how to use MicroPython to carry out actions based on the PIR sensor. The code can make decisions and set an output. This can be used to turn on LEDs. You'll also learn about schematic circuit diagrams, making it easier to understand how a circuit works.

You'll Need

- Breadboard **magpi.cc/breadboardhalf**
- PIR motion sensor **magpi.cc/pir**
- White LEDs **magpi.cc/whiteled**
- 150 Ω resistors **magpi.cc/14wresistor**
- Jumper wires **magpi.cc/jumperbumper**

01 What is a microcontroller?

A microcontroller is an electronic device. It is a type of computer, in that it runs code to determine what actions to take, but it isn't as powerful as your Raspberry Pi computer and does not run an operating system. The microcontroller runs your code directly and is great for using as part of an electronic circuit.

You'll need a Raspberry Pi Pico for this project. Pico uses the RP2040 microcontroller. It's a low-cost microcontroller packed full of features. There are versions of the Pico available with wireless connectivity, but the standard Raspberry Pi Pico is fine for this project.

02 Getting started with the Pico

To be able to insert the Pico onto a breadboard you'll need pins soldered to it. You can solder headers on yourself, or you can buy versions of the Pico with the pins pre-soldered. With the headers in place, you can plug the Pico into a breadboard ready for adding other components.

You'll also need to install MicroPython onto the Pico. Download the appropriate UF2 file from **magpi.cc/micropython**. Hold down the BOOTSEL button whilst connecting the Pico to your computer. Drag and drop the UF2 file to the RPI-RP2 drive that appears on your computer.

03 The PIR sensor and the GPIO pins

A PIR sensor is a device that can detect when someone is nearby. It stands for passive infrared and is essentially a heat detector, which detects the heat from someone's body. To prevent unwanted triggering, the sensor is split into zones; it only sends a high output when it detects heat moving across the zones.

The PIR sensor will be connected to one of the GPIO (general-purpose input/output) pins on

the Pico. These are sometimes referred to as GP (general-purpose) pins. The GPIO port numbers do not line up with the physical pin number. See the Pico pinout diagram (**Figure 1**, overleaf) for details.

04 Connect the PIR sensor

The Pico uses 3.3V for the GPIO pins. The PIR sensor is connected to the USB voltage which is approximately 5V, but fortunately the PIR's high output signal is only 3V, which is safe to use with the Pico.

The PIR sensor normally comes with male headers. You will need to use male-to-female jumper leads to connect from the breadboard to the PIR sensor. Connect the pin marked GND (ground) on the PIR to the breadboard row

> "The PIR sensor normally comes with male headers"

connected to pin 3 on the Pico, the VCC (power) pin to pin 40 on the Pico, and the OUT pin to pin 4 on the Pico. In the diagram (**Figure 2**), we've used the horizontal power rails of the breadboard, which makes adding additional components easier.

05 Adding LEDs

The aim in this project is to create light when the PIR sensor detects someone in the room. Ideally this will turn on a bright light, but that would need more power than the GPIO ports can provide. So, for now you can use multiple white LEDs.

One thing to be aware of is that the GPIO pins can only provide a small current. According to the RP2040 data sheet, the total across all the GPIO pins is 50mA. The highest current mode for an individual pin is 12mA. This is not the absolute maximum, but the output voltage will decrease when exceeding that current.

06 Limiting the current through the LEDs

This circuit uses four LEDs; limiting them to 12mA will be just within the maximum for the individual pins and the total for all pins. Using white LEDs, they typically drop 3.3V across them; with a 5V power supply, then, we need a suitable resistor to limit the current to 12mA.

Using a maximum of 1.7V dropped across the resistor and limiting to 12mA gives 141Ω, which rounds up to 150Ω. These should be wired as shown in **Figure 1** over the page.

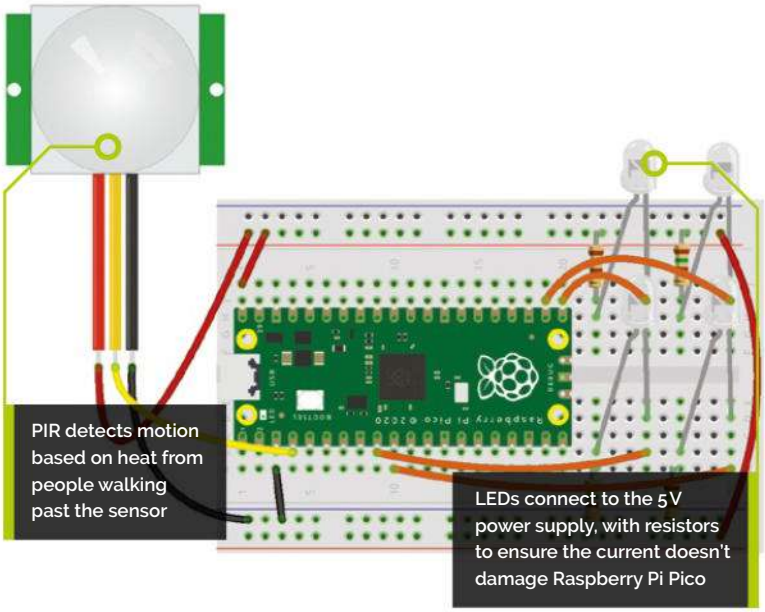


Figure 2 The breadboard wiring diagram for the project

07 Schematic circuit diagram

One of the challenges with the breadboard diagram is that as more components are added, it becomes harder to see where each of the device legs and wires go.

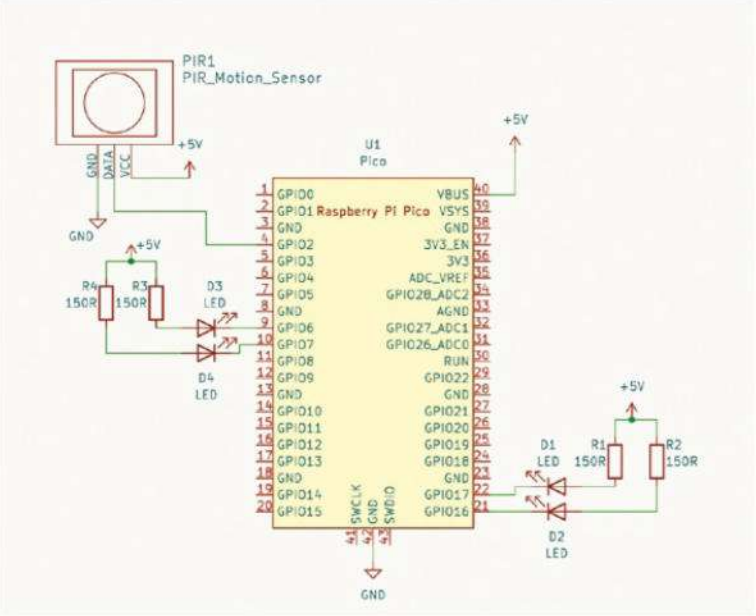


Figure 3 A schematic circuit diagram does not look the same as how it does on a breadboard, but makes it easier to see which wires connect where

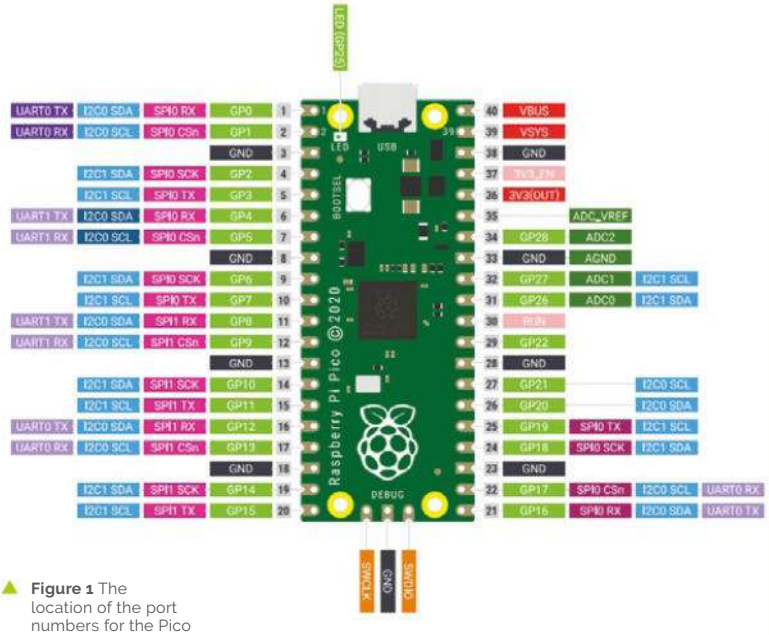


Figure 1 The location of the port numbers for the Pico are shown in this diagram. Many of these can be used for different functions

There is an alternative way of showing the circuit which is the schematic diagram. It involves learning some new symbols and conventions, but makes it much easier when creating larger circuits.

The schematic diagram for this circuit is shown in **Figure 3** on the previous page.

08 Understanding the schematic diagram

The schematic diagram includes a combination of boxes and other symbols which represent the components (see **Figure 4**). The components are then joined by lines representing a wire or electrical connection.

Raspberry Pi Pico is represented by the large central rectangle, which is a common way of representing integrated circuits or more complex modules. The labels around the outside represent the physical pin numbers, while the labels inside the box indicate the purpose, such as the GPIO port numbers. The PIR sensor is also a rectangle, which in this case includes an image inside to represent the sensor.

The resistors are represented by small rectangles. The LEDs are made up of a diode, which is an arrow followed by a vertical line, and then two arrows which indicate that the LED emits light. The line indicates the cathode (negative end of the diode).



Select the Pico and port from the Configure Interpreter menu option. This is how the Pico appears when connected to a Raspberry Pi

09 Power connections

The power supply connections are shown through symbols. The arrows pointing upwards are used to indicate the positive power connection which is connected to 5V from the Pico. These are all connected.

The ground, or negative, connection is shown as a small diagram with the label GND. They are all connected.

From the diagram, you can see the pins that are used on the Pico and how all of the components are wired together. For complex circuits, this is much easier than trying to follow the wires on the breadboard diagram.

10 Start programming

To program Raspberry Pi Pico, you'll first need to connect using Thonny. It may connect automatically. If not, then from the Run menu choose 'Select interpreter' and choose 'MicroPython (Raspberry Pi Pico)'. Select the detected port; if running Thonny on a Raspberry Pi, it'll normally say 'Board in FS mode'; for Windows, it'll be the appropriate USB serial port.

You can write the code in the main window and save it to the Pico. If you name the file **main.py** then it will run automatically when your Pico is powered on – any other file name and you will need to run it manually using the Thonny editor.

11 Control through code

The code starts by importing the relevant modules – the machine module is a standard MicroPython library for handling GPIO on microcontrollers, and the Pin class is used to represent an individual GPIO pin. The utime module is the MicroPython version of the standard Python time module. It's used to add delays in the code.

THE MAGPI



This tutorial is from in The MagPi, the official Raspberry Pi magazine. Each issue includes a huge variety of projects, tutorials, tips and tricks to help you get the most out of your Raspberry Pi. Find out more at **magpi.cc**

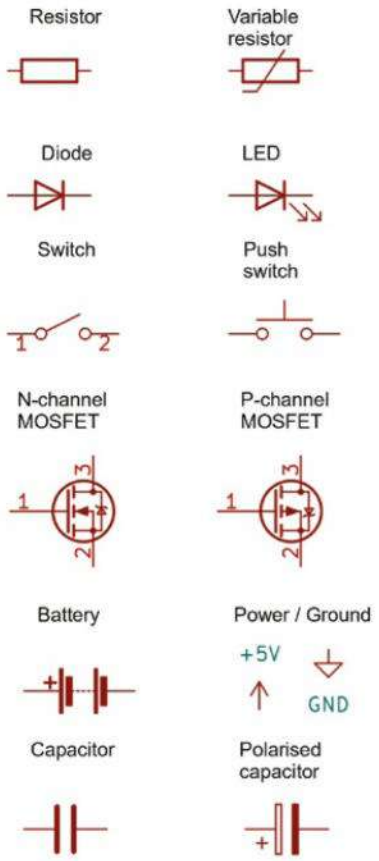


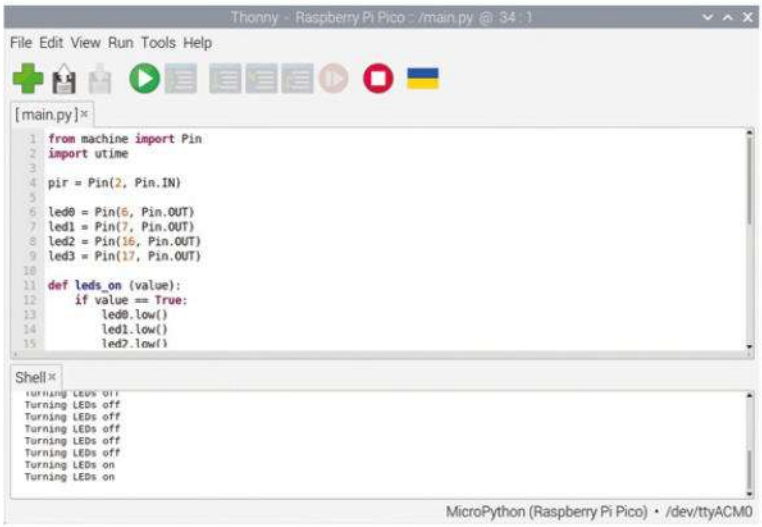
Figure 4 Some common circuit symbols used on schematic diagrams. There are different standards – these are based on symbols used by the KiCad circuit designer

The PIR sensor is used as an input and the four pins for the LEDs are defined as outputs.

There is a function called `leds_on()` which can be passed either True or False as an argument. If True is passed, then it turns all the LEDs on; if False is passed, then it turns all the LEDs off. Note that the pin is set to low to turn an LED on, and high to turn it off. This may appear counter-intuitive, but is because the LEDs are connected to the positive supply and the GPIO needs to be low to allow the current to flow into Pico.

12 Detecting the PIR value

The rest of the code is contained within a `while True:` loop, which will constantly repeat. If the PIR gives a high signal to the pin, the value is set to 1 and the LEDs turn on. If it has a low signal, then the value is 0. There is a delay in each of the conditions – it is intentionally longer when the LEDs are turned on to ensure it stays on for a reasonable period of time when triggered. In this case it stays on for ten seconds, but if the PIR does not detect anyone, then it checks every second.



The Thonny editor can be used to write MicroPython code directly onto the Pico. Switch to standard mode to see all the options

main.py

DOWNLOAD THE FULL CODE:
magpi.cc/picopir

Language: **MicroPython**

```
001. from machine import Pin
002. import utime
003.
004. pir = Pin(2, Pin.IN)
005.
006. led0 = Pin(6, Pin.OUT)
007. led1 = Pin(7, Pin.OUT)
008. led2 = Pin(16, Pin.OUT)
009. led3 = Pin(17, Pin.OUT)
010.
011. def leds_on (value):
012.     if value == True:
013.         led0.low()
014.         led1.low()
015.         led2.low()
016.         led3.low()
017.     else:
018.         led0.high()
019.         led1.high()
020.         led2.high()
021.         led3.high()
022.
023. while True:
024.     if (pir.value() == 1):
025.         print ("Turning LEDs on")
026.         leds_on(True)
027.         utime.sleep (10)
028.     else:
029.         print ("Turning LEDs off")
030.         leds_on(False)
031.         utime.sleep (1)
```

Top Tip

Pico numbering

With the USB connector positioned to the left, the pins are numbered starting at 1 in the bottom left. Along the bottom row up to pin 20, then from right to left along the top row from 21 to 40.

3D-printing polycarbonate

When the going gets tough, the tough get PC



Ben Everard

@ben_everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

When it comes to printing polycarbonate (PC), there's one basic problem – warping. Warping happens because as the 3D print cools, it shrinks. Different plastics shrink in varying amounts as they cool, but the problem with polycarbonate is not that it shrinks a lot per degree, but that you have to print it very hot, so it's got a long way to cool.

You generally need a nozzle temperature of around 300°C. The bed temperature should be somewhere around or over 100°C. In most cases, hobbyist printers don't have heated chambers, but at the very least, you're going to need an enclosure. The aim of the game with polycarbonate is to slow down its cooling, because slow cooling means more even cooling, and more even cooling means less warping. Whatever you can do to help here will make a difference. How big the problem will be depends on the model. Wide, flat models tend to warp quite a bit, while small models don't warp so much.

A slightly ironic problem is that polycarbonate tends to be used for mechanical parts, and mechanical parts usually have to fit together, so warping is both more likely and disproportionately a problem.

The second problem is the temperature. Polycarbonate prints hot. There are some blends that will print at around 290°C, but many need 300°C or higher. How high can your printer go? The answer here is usually not what temperature your printer could possibly reach, but how hot should you get it to go. Most printers will have a heater that can reach these temperatures, however, they also have parts that may become damaged if you try to heat it up to this temperature.

A rough anatomy of a 3D printer hot end is that there's a block of aluminium called a heater block. A heater and thermistor are placed in holes in this, and there's a tapped hole for the nozzle. The problem now is how to mount this on a (usually plastic) extruder without everything melting. The solution is a heat break, which is a metal tube, usually with fins on it and a fan blowing over it. This should hopefully stop heat creeping up into the rest of the machine. However, in order to help the filament go down the middle of the heat break, there's usually a PTFE (Teflon)



Right
Small features, like Benchy's chimney, are prone to distortion if they don't have time to cool sufficiently

tube down the middle. This PTFE will break down at high temperatures. How high you should push your printer is then determined by how well the heat break works, and how willing you are to risk damaging your PTFE tube (it can give off highly toxic gases if it gets too hot). Not all printers have this PTFE tube – printers that don't are usually advertised as having an 'all-metal hot end'.

It is, however, not only the enclosure and the nozzle that you have to get up to temperature; you also need a heated bed that can get up to around 110°C.

The final piece of the puzzle is getting it to stick (or not stick) to the print bed. Using a glue stick on a textured PEI sheet is ideal for this.

There are a lot of different blends of polycarbonate, many of them designed to be easy to print, but in the world of polycarbonate, easy is a relative word. Because of this variation, it's difficult to give hard and fast rules for print settings. Experiment and see what works with your setup and your filament. You'll most probably need a few goes before you dial in your settings.

That's an awful lot of hoops to jump through in order to print polycarbonate, but there are some really good reasons to put yourself through all of that. The first is durability. Polycarbonate parts are great if you need strong parts that are also durable, particularly if they're mechanical.



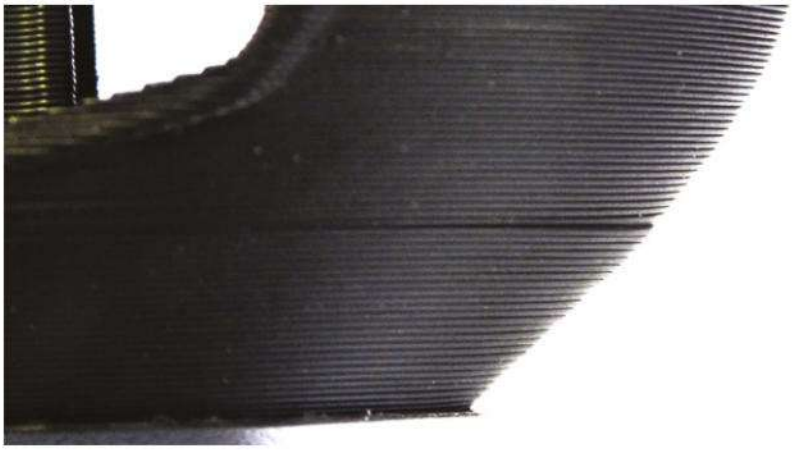
It's this performance that leads polycarbonate to be referred to as an 'engineering' filament. While it is a bit tricky to print, it's still easier and cheaper than milling metal.

Another big advantage is its temperature stability. 3D-printable plastics, by their very nature, soften in the heat. The high-temperature requirements for printing polycarbonate translate directly into its ability to keep its strength when exposed to heat.

> While it is a bit tricky to print, **it's still easier and cheaper than milling metal**

Polycarbonate is rarely a filament you reach for because you want to print with it. It's a filament you reach for when the other filaments in your armoury don't have the properties you need. With a bit of coaxing, it is printable on some hobbyist printers, but be prepared to take a bit of time over it.

Above
Polycarbonate pushes hobbyist printers to their limits, but if you pick the right blend, it might be possible



Below
Polycarbonate can have excellent layer adhesion, but warping can cause separation if the heat isn't well managed

DRAFT SHIELD

When printing filaments that are prone to warping, one option is to print a 'draft shield', which is a one-perimeter-thick surround a couple of centimetres outside your main print. The idea is that this acts as a sort of mini enclosure and holds some of the heat in. It's obviously not as effective as an enclosure, but could be used alongside one.

However, while we've had success using these with some filaments (such as ASA), we didn't find it reliable with polycarbonate. It tended to disintegrate into spaghetti; however, this might just be us. If you are having problems with warping, it might be worth investigating.

In PrusaSlicer, you can find the option to enable this under Print Settings > Skirt And Brim (you'll need to have Advanced or Expert mode enabled).

TUTORIAL

Pico W-powered Bluetooth chord keyboard

Make a chord keyboard that works over Bluetooth



Rob Miles

@robmiles

Rob Miles has been playing with hardware and software since almost before there was hardware and software. You can find out more about his so-called life at **robmiles.com**.

Figure 1 The key positions have been customised for the author's hands. His left hand is slightly larger than his right

A **chord keyboard is a great way to enter text on the move.** You can type in text without having to look at your fingers. This project builds on the chord keyboard described in Issue 46 of HackSpace magazine, adding Bluetooth connectivity, using a custom PCB, and making it easy to create a keyboard that matches your hand dimensions.

THE PICO 'BLUE CHORDS' KEYBOARD

Figure 1 shows left- and right-handed versions of the keyboard. They can be used to enter text, numbers, and symbols. The OLED screen displays the keyboard mode and the last few characters that were entered, along with the Bluetooth connection status. The keyboard is powered by a Raspberry Pi Pico W running C++ code. A rechargeable battery makes the keyboard fully portable. The project uses the KiCad circuit design tool to create the custom-printed circuit board (PCB) and the FreeCAD 3D modeller to create the case. The positions of the keys and the case layouts are synchronised by Python code running inside KiCad and FreeCAD, allowing for flexible key positioning and ensuring that the holes for the keys match the key positions on the circuit board. You can find the designs, along with the C++ source for the keyboard, on the GitHub repository for this project here: **github.com/CrazyRobMiles/PICO-blue-chords**.



TYPING WITH CHORDS

You can see the chords used to enter text using the keyboard in **Figure 2**. The chord system used by this keyboard was first used on devices produced by the Microwriter company in the 1980s. There are six keys: one for each finger and two for the thumb. You enter characters by pressing combinations of keys. **Figure 2** shows the chords for the alphabet. The keyboard understands chords for text, numbers, and a range of symbols. Each of these is selected by putting the keyboard into one of four modes. You select the mode of the keyboard using the 'control' key at the far left. The thumb moves to this key to create chords to change the mode. The mode is displayed at the bottom of the OLED screen. Documentation for the keyboard and charts for all the chord combinations are on the GitHub site for this project. With a bit of practice, you can enter text surprisingly quickly.

START WITH THE CIRCUIT

Figure 3 shows the circuit design that was created using KiCad. There are multiple connections for the displays and the keys, so the same board can be used for left-handed and right-handed configurations.

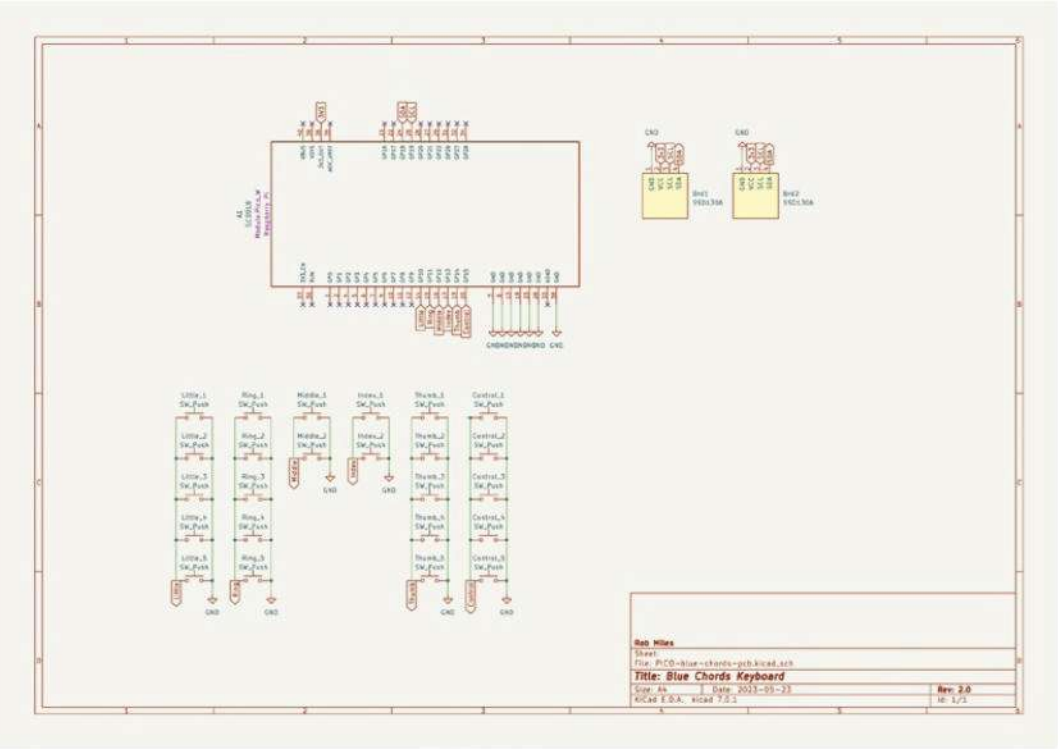


Figure 3
Only one switch will be placed in each column

The circuit has no connections for power supply and battery management. A LiPo SHIM for Pico from Pimoroni (**pimoroni.com**) was used for these. Once the design for the circuit was complete, it was time to lay out the PCB.

> The same board can be used for **left-handed and right-handed configuration**

PCB NUMBER ONE

Figure 4, overleaf, shows the PCB design that was sent to be manufactured. This was the first PCB the author had designed from scratch. It turned out to be quite an easy process. You swap between the Schematic Editor (where you design your circuit) and the PCB Editor (where you lay out your PCB) as you refine the design. For the author, the hardest task was finding the right 'footprint' designs for each component. The footprints tell the PCB design program what a component looks like on the PCB. For example, the circuit diagram contains a 'push →

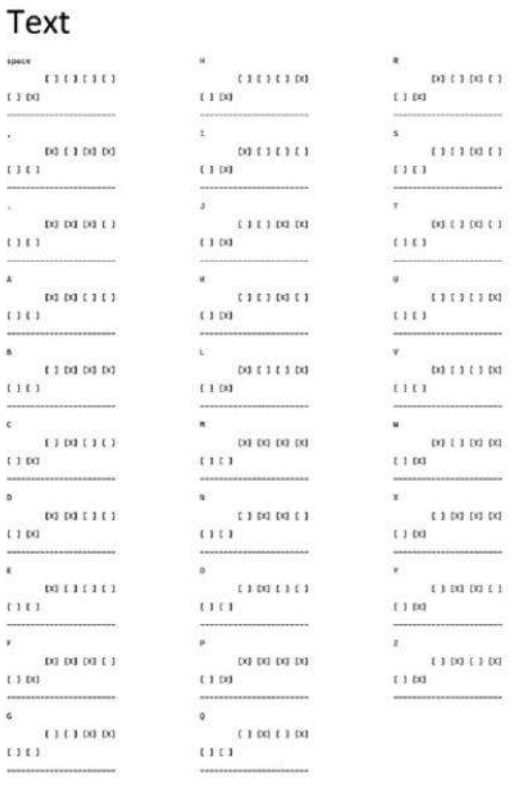


Figure 2
The chords are shown for right-handed use. They are reversed for left-handed use

YOU'LL NEED

- **PCB**
 There is a custom-printed circuit board (PCB) design available for this project, although you can connect the circuit by hand if you prefer
- **Six key switches**
 The Kailh mechanical keyboard switches work well
- **Six Kailh switch sockets for MX-compatible mechanical keys**
 These are soldered onto the PCB
- **Six keycaps**
- **An SSD1306 OLED display**
 Make sure that you get the monochrome version
- **A Raspberry Pi Pico W**
- **A micro USB cable to link the Pico W to the host**
 You will need this for programming and charging
- **A battery and power supply**
 The author used the LiPo SHIM for Pico from Pimoroni along with a Galleon 400mAh hard case LiPo battery
- **A case**
 There is a 3D-printable design, or you can put your keyboard in any box you like

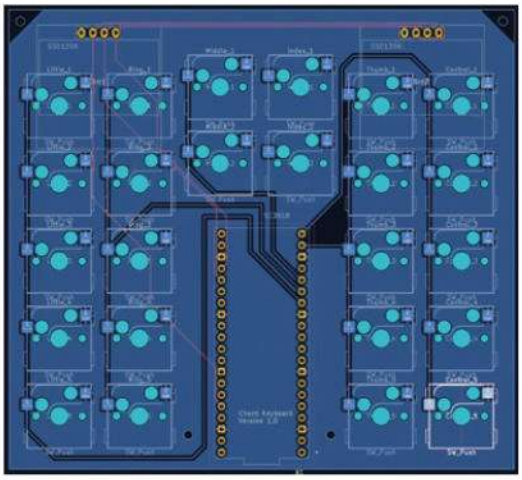


Figure 4
Can you spot which column of keys was not connected? The author didn't…

button switch', but the switch footprint required was called 'SW_Hotswap_Kailh_MX'. This is the footprint for a socket into which the keys are fitted. The socket goes on the reverse of the board and the key connects through the board to it. The socket has the characteristic shape you can see above on the diagram, with solder pads for the socket connectors. This footprint file for this design had to be located and added to a library in the project. Then each switch in the circuit diagram was associated with that footprint style. A similar process was followed to get the footprint for the Pico device. There are also 3D designs for most components, so you can use these to make a 3D visualisation of the board. The author was just happy that he managed to get components that looked as if they should work.

GETTING THE BOARD MADE WAS THE EASIEST PART

The author was expecting all kinds of problems getting the circuit board made, but this was the easiest part of the process. You can add 'plugin' elements to KiCad, and one of the things you can add is a plugin from the PCB manufacturer. In the author's case, this was PCBWay (**pcbway.com**), which had generously offered to sponsor the project. The plugin prepared the project for submission, sent it to the manufacturer, and then opened a web page for the order. The page also showed a visualisation of the board for checking. The process was all very smooth and well-managed – six days after the order was submitted, the finished boards arrived.

Figure 5 shows a completed keyboard. The switches have been placed for left-handed use. The thumb is used to select between text entry and control functions. The author had hopes of being able to use a chord keyboard with his left hand and the mouse with his right. He was hoping that his brain would transpose chord designs learnt for the right hand onto the left hand. This was interesting in practice. It turns out that the author's brain can perform this kind of mapping, but it only works if he doesn't think about what he is doing. As soon as he thinks about the keys he needs to press to enter an 'A' with his left hand, he discovers that he is unable to do it. Hopefully this will improve with practice.

NOTHING IS EVER PERFECT

Figure 6 shows the rear of the PCB. The production process did a perfect job of converting the designs into a board, but the design was flawed. The extra wires are needed because the author missed off the connection for one column of sockets and wired a display the wrong way round. However, he thinks that it is traditional that the first design of any board should contain mistakes and is proud to be upholding this.

LIPO SHIM FOR PICO

Given the author's prowess with circuit design, you will be relieved to discover that the PCB he designed doesn't do any of the potentially dangerous power supply management. This is performed by a LiPo SHIM for Pico from Pimoroni, which you can see at the bottom of the PCB in **Figure 6**. The SHIM fits over pins on the Pico and connects to a rechargeable battery. It provides an on-off switch, white power on LED, and red charging LED. It is soldered onto the pins from the Pico which fit through the PCB, enabling a slimline construction. The base of the Pico exposes the micro USB socket, which is used

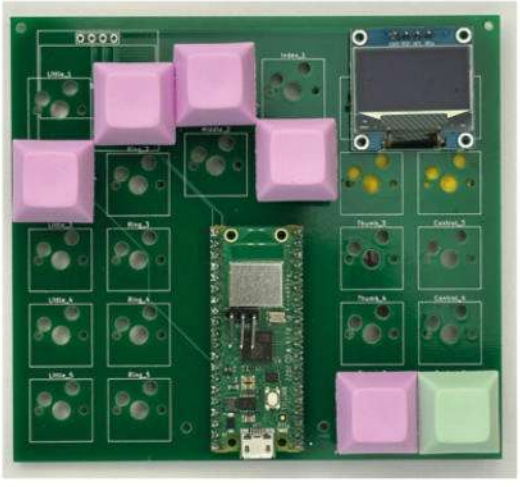


Figure 5
The tracks on the top of the board connect the Pico to the display. Tracks on the bottom connect the key switches

QUICK TIP

Rather than add wires to your circuit diagrams, you can label inputs and outputs and KiCad will connect them. If the author had done this, he wouldn't have got the connections the wrong way round on one of the displays.

for charging. The SHIM exposes the indicator LEDs and power button. The author has found this to be a neat way to create battery-powered Pico projects.

PANEL BEATING

Figure 7 shows the front panel of the keyboard. It is a solid piece of printing which has deeply recessed holes for each of the keys. The positions of the keyholes must exactly line up with the sockets in the PCB.

Figure 8, overleaf, shows the case being designed in FreeCAD. On the right-hand side, you can see the Python code that is used to create objects that describe the positions of the keys on the PCB.

```
Index_1=["Index_1",168.60,52.66,17.38,17.38]
```

The statement above shows the object that defines the position of the hole for the first index finger key. The first item in the object is the name of the key, this is followed by the X and Y coordinates and the width and height of the hole. The variable **keyHoles** at the bottom of the code contains a list of the key objects to be used for this keyboard. It also sets the position of the hole for the display.

```
keyHoles = [Little_5, Ring_5, Middle_2, Index_1,
Thumb_1, Control_2]
```

If you want a different set of key holes, you just change the contents of this list. The hole information is extracted from the PCB design by another piece of Python that runs inside FreeCAD and generates these object positions:

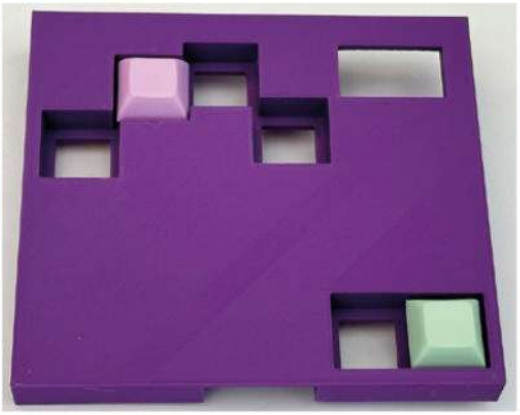


Figure 7
The keys snap into the holes in the front panel. They are not supported by the PCB underneath. They just plug into it

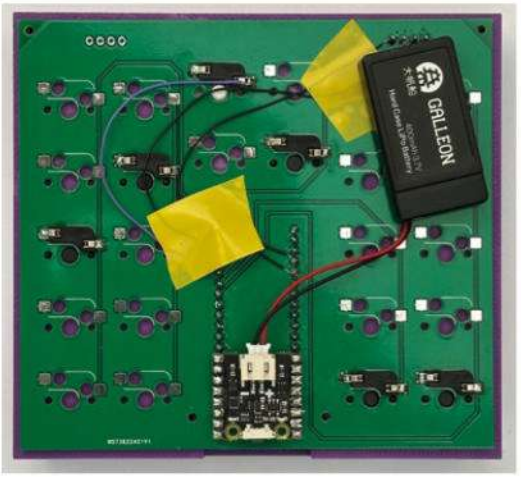


Figure 6
If you want to be able to change the key configuration, you can add extra switch sockets for the additional key positions that you want to use

```
import pcbnew

file = "C:\\Users\\rsmil\\Blue Chord\\pcbKeyboard.
KiCad_pcb"

board = pcbnew.LoadBoard(file)

l = []

for f in board.GetFootprints():
    b = f.GetBoundingBox()
    ref = f.GetReference()
    p = f.GetPosition()
    px = (p[0] / 1000000.0)
    py = (p[1] / 1000000.0)
    height = b.GetHeight() / 1000000.0
    width = b.GetWidth() / 1000000.0
    t = '{0} = ["{0}",{1:.2f},{2:.2f},{3:.2f},{3:
.2f}]'.format(ref,px,py,height,width)
    l.append(t)

l.sort()
for t in l:
    print(t)
```

The Python code above runs inside the KiCad PCB Editor and uses the pcbnew library to read the footprint positions from the PCB design. It works through the footprints, extracting the name and position of each and assembling an object description. Each of the definitions is added to a list which is then sorted and printed. The program uses the Python format feature to print out all the values in millimetres with only two decimal places. This provides the list of component positions you can see in FreeCAD in **Figure 8** and **Figure 9**, overleaf.

QUICK TIP

Don't be afraid to add a bit of weight to your prints. The keyboard is all the better for being heavier; it is less inclined to move about when used. The author considers that the extra heft was worth the six-hour printing time.

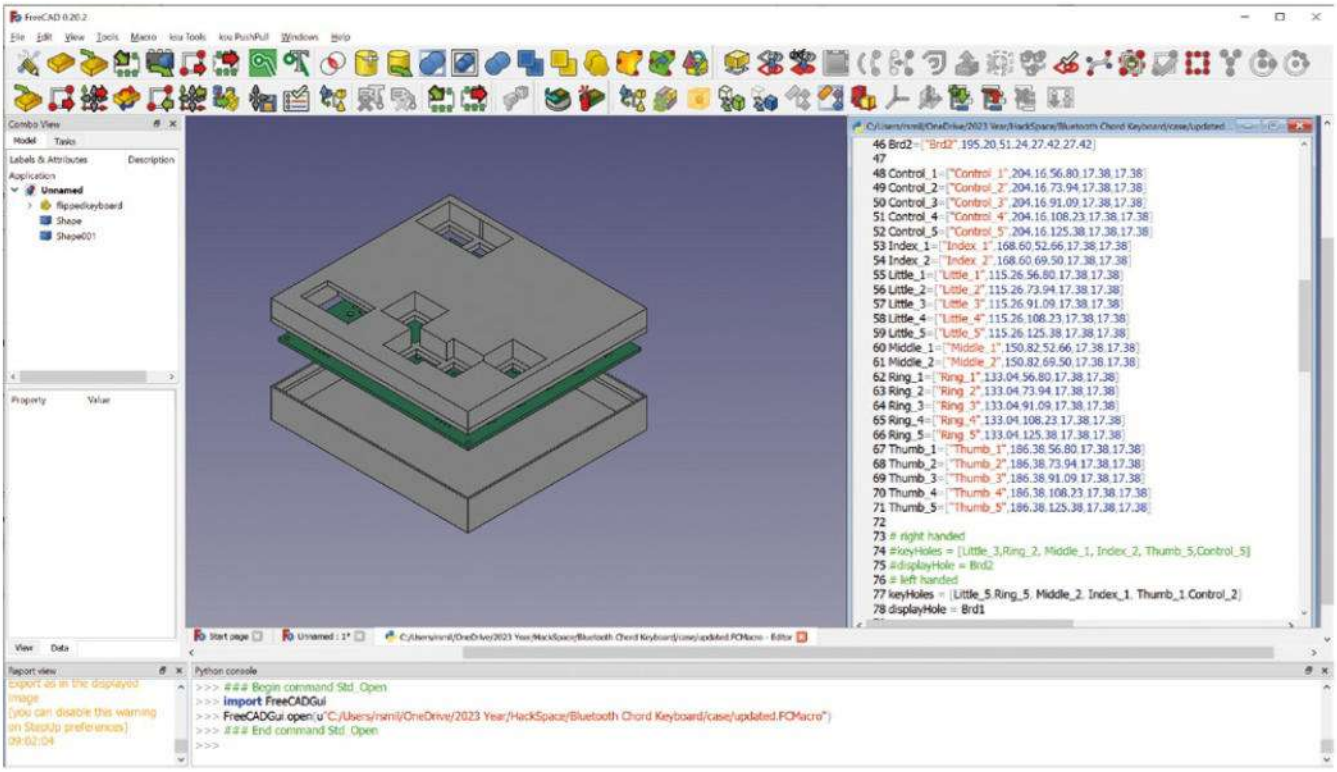


Figure 8
The kicadStepUp plugin for FreeCAD has been used to import the PCB design so that it can be checked against the case design

Figure 9
The shell window contains the output from all the print statements in the program

Figure 9 shows the Python window in KiCad with the output from the program. The amount of footprint data being transferred was very small. If you had a large amount of position information to transfer into FreeCAD, you could write the code into a file to be included in the FreeCAD script. This would allow you to closely integrate the PCB and case creation.

ONE FUNCTION TO RULE THEM ALL

The keyboard, as designed, works well, but it turns out that the keys are a bit too close together. The sides of some of the keycaps needed to be filed down a tiny amount so that they moved smoothly. The design needs an increase in the spacing of all the keys by about half a millimetre. At this point, the author has decided that it would be a good idea to use a Python program running in KiCad to position the key footprints as well as generate the position information for FreeCAD. So, the next version of the keyboard will be created using this approach.



C++-POWERED KEYBOARD

The code that controls the keyboard is based on the Pico W Bluetooth example 'hid_keyboard_demo' application from the Pico examples at **github.com/raspberrypi/pico-examples**. This program has a demonstration mode which sends a preset string of text out of the keyboard. It uses a timer to trigger the transmission of each character. The timer has been repurposed to call the **scan_keys** function, as shown below:

```
#define SCAN_INTERVAL_MS 20

static void key_scan_timer_handler(btstack_timer_
source_t *ts)
{
    UNUSED(ts);
    // scan the keyboard
    char ch = scan_keys();
    // do we have a key press?
```

```
    if (ch != 0)
    {
        // send the key
        stdin_process(ch);
    }
    // set timer for next character
    btstack_run_loop_set_timer_handler(&demo_
text_timer, key_scan_timer_handler);
    btstack_run_loop_set_timer(&demo_text_timer,
SCAN_INTERVAL_MS);
    btstack_run_loop_add_timer(&demo_text_timer);
}
```

The **key_scan_timer_handler** function is called every 20 milliseconds (50 times a second). It calls the **scan_keys** function and then restarts the timer. The **scan_keys** function performs 'debounce' on the input signals. Mechanical switches can produce a rapid sequence of high and low outputs as their contacts open and close. The debounce code ignores these short pulses. A non-zero result from **scan_keys** indicates that a character has been entered. The **stdin_process** function is called to transmit the character. The **stdin_process** function is part of the original demo code, where it is used to transmit characters arriving on the standard input stream. This function also manages Bluetooth pairing, connection, and disconnection. If you want to develop the code further, you can find it all in the project repository. The keyboard also uses a library for the SSD1306 display, which you can find on GitHub here: **github.com/daschr/pico-ssd1306**.

> "It is useful to be able to configure the key positions; **it is surprising how different one hand can be from another**"

USING CHATGPT TO CONVERT FROM PYTHON TO C++

This project is based on a previous one written in the Python language. A central component of the Python solution is a set of lookup tables that decode the chord bit patterns from the input switches into character codes. These needed to be converted from Python to C++. The author thought he might use the ChatGPT chatbot (**chat.openai.com**) to do this. ChatGPT is a language processing tool that has been trained on many millions of articles, including lots about programming. It is adept at taking language input and transforming it. So, a session was opened, and the Python code was pasted in. ChatGPT was then asked to produce a C++ version. This turned out to be quite successful. ChatGPT can take Python arrays and dictionaries and convert them into equivalent C++ code. However, the code it produces is quite long-winded, so in the end, the author converted the tables by hand using a few edits.

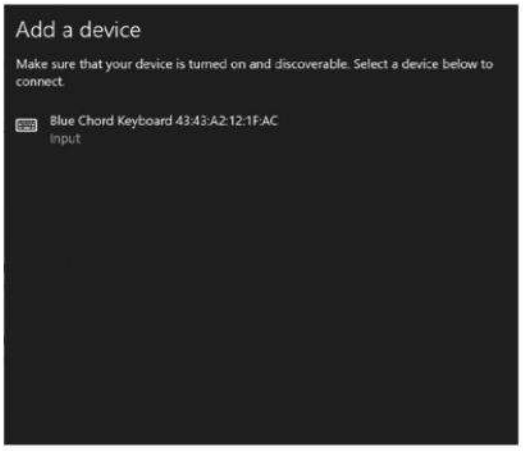


Figure 10 The keyboard generates a different Bluetooth address each time it starts

THE RE-PAIR SHOP

Before the keyboard can be used with a device, it must first be paired with it (**Figure 10**). Follow the Bluetooth pairing process for the device you want to use it with. The screenshot above is from Windows 10. Once the keyboard has been paired, it will remake a connection if the device disconnects. However, if you turn the keyboard off and on again, you may have to re-pair it. This is because the demonstration code the keyboard uses doesn't retain pairing information inside the Pico. The good news is that battery life is good, so you don't have to turn the keyboard off very often.

KEY POINTS

The keyboard was fun to make and works well. It is useful to be able to configure the key positions; it is surprising how different one hand can be from another. The author even found differences in his left and right hands, requiring different keyboard arrangements for each. The integration of Python into the build process means that you can create printable designs for cases which exactly match the PCBs. All the software for the keyboard is in the repository for this project on GitHub, so you can make your own version. There are also code images for left-handed and right-handed versions of the keyboard code that you can just copy into your Pico W if you don't want to build your own software. The author is looking forward to making more PCBs.

Planetary gears

Get a wide range of gear ratios in a small space



Ben Everard

@ben_everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

We've looked previously at how to 3D-print gears, where one gear drives another and the ratios of the two gears determine the relative speed of the two gears. This can work well, but does have a few problems. Firstly, it can take up a lot of space and, secondly, it's not the most durable system. One alternative is a planetary gear-box. In this system, there are three sets of gears: an inner (or 'sun') gear, three or more 'planetary' gears that rotate around it, and a ring gear around the outside (see **Figure 1**).

There are three potential inputs into this system – through the sun gear, through an attachment to the planetary gears as they rotate around the sun, and through the rotation to the ring gear. In order for the gear system to be useful, one of these has to be held still, one has to be driven, and the final one is the output.

The ratios and the orientation of the output will depend on not only the sizes of the gears, but also the way in which the gears are driven (or held).

What does all this mean in practice? Let's take a look with a simple model downloadable from: **printables.com/model/495872**.

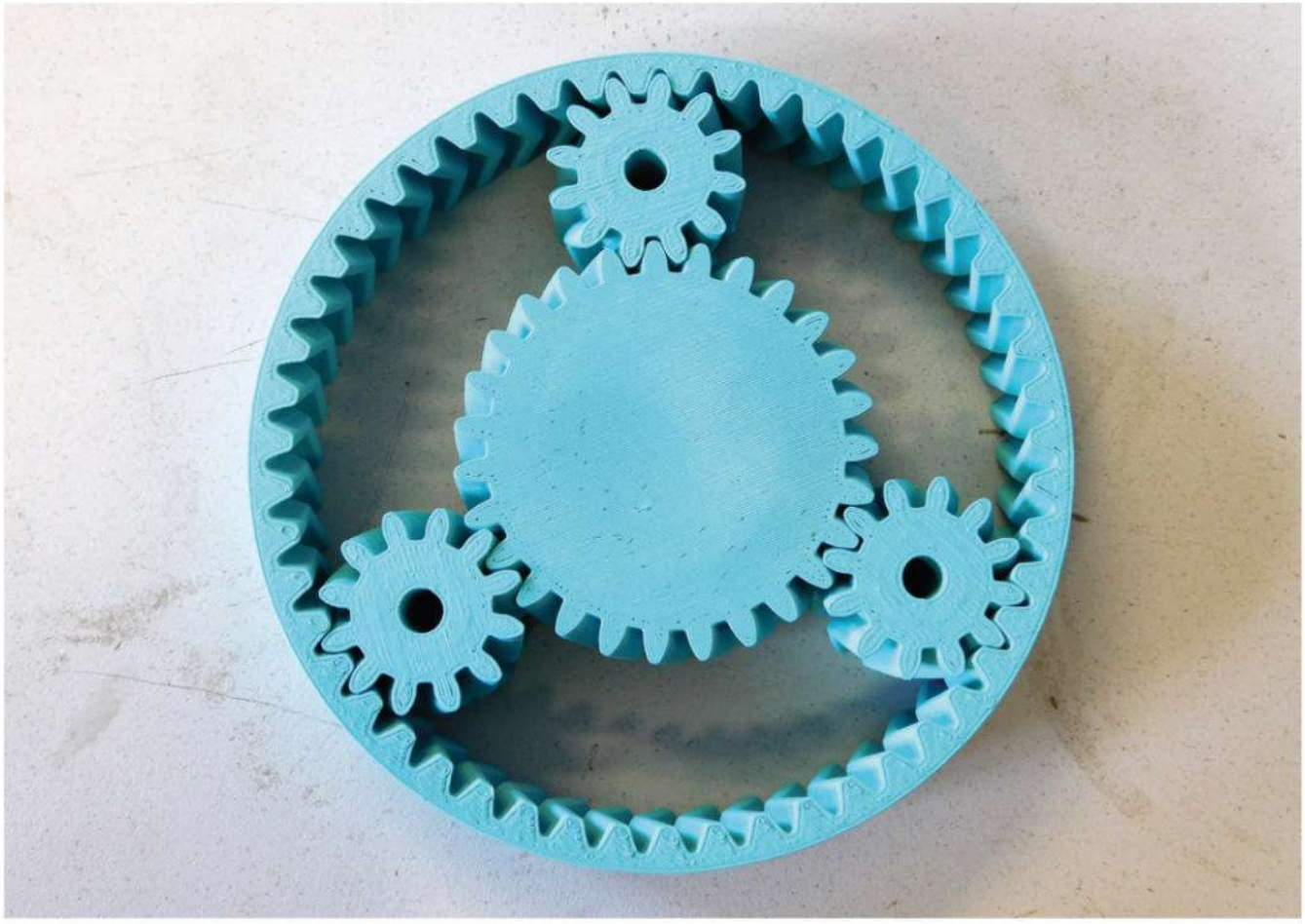
There are three planetary gears, one sun gear, and one ring gear. These all have herringbone teeth which help them stay in place without any internal supports, but these do mean that it's a bit of a squeeze to put everything together. The planetary gears do have to be positioned quite precisely for the whole thing to work, but we've found the easiest option is to jam them in at roughly the right place, and then adjust them later.

Once you've got all the bits in place, the first thing to do is play with it. It's ridiculously addictive to just spin this in your fingers in different ways, and it's probably worth printing just for this alone. However, it's a bit hard to really get a feel for how it works by just playing with it. This is because there are three points of control and humans only have, at most, two hands. In order for us to gain a bit more out of it, we're going to need some extra bits.

> "It's ridiculously addictive to just spin this in **your fingers in different ways**"

The first of these is the planet assembly. This is an attachment, with axles for each of the planetary gears, that allows you to extract the rotation from them. To put it in, simply put each of the axles into the centre of a planetary gear. They should fit in





perfectly – if they don't, then the gear isn't in quite the right position. To get it into position, bend the ring gear slightly so it disengages, then rotate the planet a little, and re-engage the ring gear. Repeat this until all three planets fit perfectly into the assembly.

Now, if you hold the ring gear still, and rotate the sun gear, the planetary assembly should spin. The crucial part of a gear chain, though, isn't that it spins, it's that it spins the correct amount.

Calculating the gear ratios of a planetary gear system is a little more complex than with regular gears (but only slightly), so you might have to break out the calculator.

We'll be looking at the transmission ratio, which is the number of times the drive gear turns for each turn of the output gear. A transmission ratio of less than 1 is a reduction, meaning that the output spins less than the drive (but will have more torque), whereas a transmission ratio of more than 1 means →

3D-PRINTABLE PLANETARY THINGS

If you're looking to explore planetary gears further, there are some more fun things you can print.

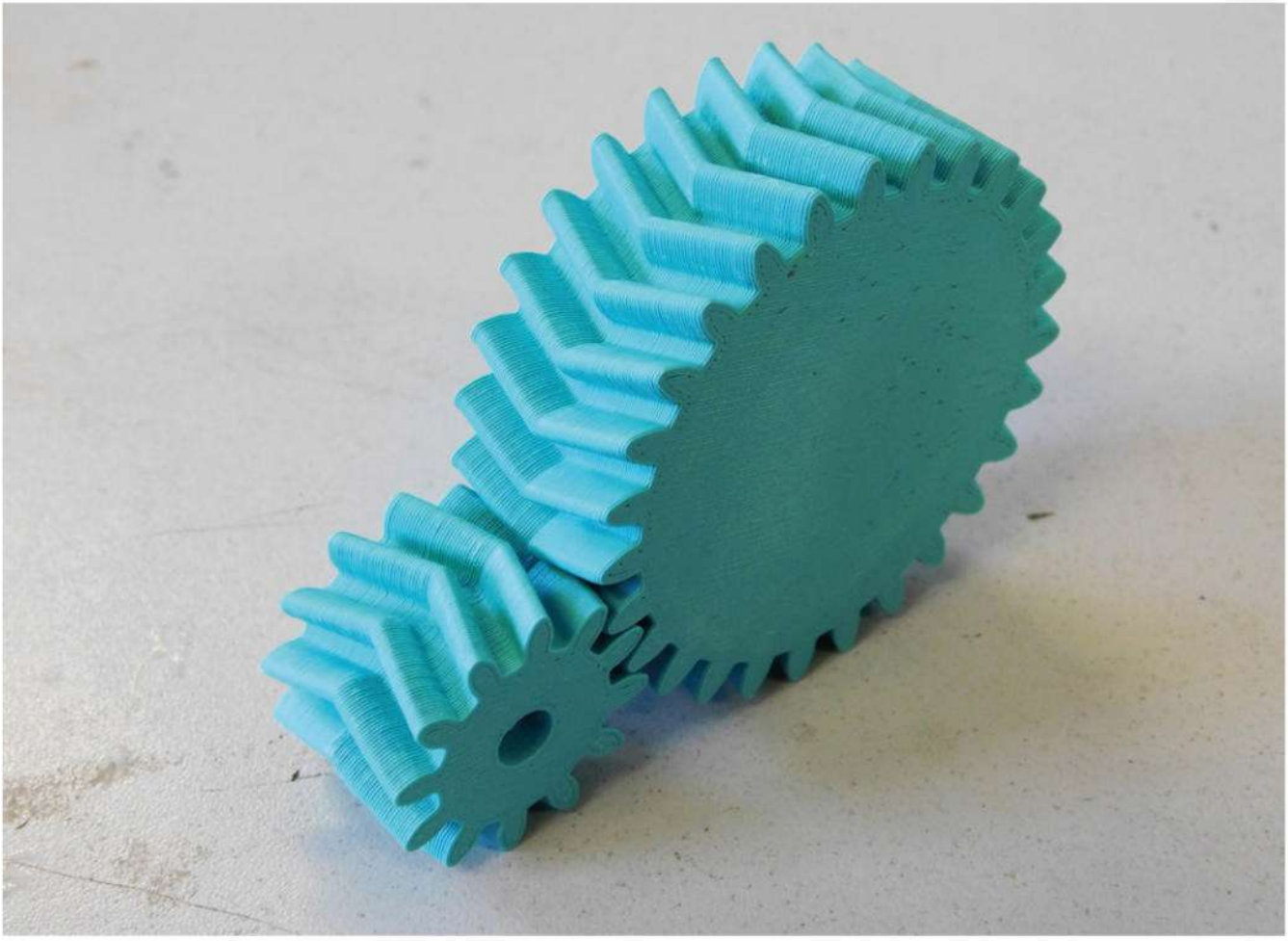
There's a little toy at **hsmag.cc/PlanetaryGear**

Or you can add power to your build with this stepper motor gear box **hsmag.cc/Nema17**

This example uses planets of different sizes to create a motion where the sun gear moves around while it spins. This spirograph-like motion can be used to stir ground coffee in an espresso portafilter. Does it make better coffee? No idea, but it looks cool – **hsmag.cc/Umikot**

In this print, the planet carrier is the hour hand, the sun is the minute hand, and the ring is fixed – **hsmag.cc/SelfadjustingClock**

Figure 1
There's something beautiful about the planetary arrangement



Above
The gears have to be the correct way around to mesh

THREE-SPEED HUBS

For simple systems, it might be enough to use planetary gears in fixed ratios. However, in more complex systems, you might want to change the ratio. The most famous variable-speed planetary gearbox is the 'Sturmey-Archer' system that provides three speeds and can be embedded directly in the hub of a bike wheel (not to be confused with the more common external style of derailleur gears).

In this system, the power is driven through either the planetary gears or the ring gears, and the power output is also from these two. Depending on which way around the drive is (either driven through the ring and output from the planets, or driven from the planets and output from the ring), you get high or low gearing from the hub. The middle gear is direct-drive.

These were first devised in 1902, and are still in use today because they can be significantly lower maintenance than external bike gears.

that the output spins more than the drive (but with less torque). In this case (with the ring gear held still), you can use the following formula:

Transmission ratio = 1 + (number of teeth on the ring gear / number of teeth on the sun gear)

In our case, there are 51 teeth on the ring gear and 27 teeth on the sun gear. This means that the transmission ratio (or the number of times the sun gear rotates for every rotation of the carrier gear) is 2.9. What's more, we've achieved this reduction in a small space with the same axis of rotation.

That isn't the only way you can transmit power through these gears, though. If you print the buggy body, you can place this on the assembly to keep the planetary gears in the same place. This is designed to be part of a wheeled buggy, where the ring is the wheel itself and it's driven through the sun gear. In



this case, the planets are held stationary.

Transmission ratio = -(number of teeth on the ring gear / number of teeth on the sun gear).

Here, the ratio is -1.9. It's negative because the direction of rotation is reversed. This method of driving wheels gives us much more torque than we would otherwise have. Though, if you do want to do this on a real buggy, you'll need to make the wheels far stronger than our test gears because they easily deform and stick.

> **The ratios here are obviously very dependent** on the numbers of teeth on the gears

There is, as you might expect, a third way of driving this gear. You can hold the sun gear stationary, and drive either the ring gear or the planetary carrier. This is harder to do with our setup, but not impossible. The transmission ratio between the ring and the planetary carrier, in this case, is calculated using the equation:

Transmission ratio = 1+(number of teeth on the sun gear / number of teeth on the ring gear)

In our case, this is 1.53. This gives us three different ratios we can achieve. There is actually a fourth option – if you fix the planetary carrier to the ring gear, then you get a direct-drive system with a ratio of 1.

The ratios here are obviously very dependent on the number of teeth on the gears. However, you can't just mash together any three different-sized gears and hope that you'll have a planetary gear system. Everything needs to fit together, and that means that it has to satisfy the equation:

(Number of teeth on the sun gear + number of teeth on the ring gear) / number of planetary gears = a whole number.

In this case, we're using a sun gear with 27 teeth, a ring gear with 51 teeth and three planetary gears, so this condition is satisfied. You can calculate the number of teeth on a gear using:

Number of teeth on the ring gear = 2 × (number of teeth on the planetary gear + number of teeth on the sun gear).

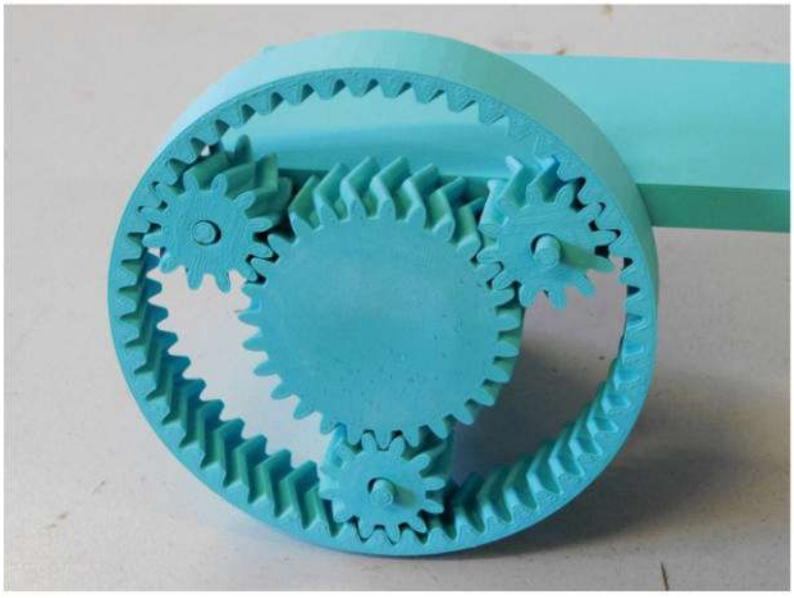
You can rearrange this to calculate the number of teeth on the third type of gear once you've fixed the number of teeth on the other two. In our case, we know our ring gear has 51 teeth and the planetary gear has 27 teeth, so, we can rearrange this to:

Number of teeth on the planetary gear = (number of teeth on the ring gear – number of teeth on the sun gear) / 2

This gives us twelve teeth on the planetary gear. Planetary gears can be a great way of getting the ratios we need in a small size. The total size of the gear system can be the size of just a single traditional gear. 3D-printable herringbone patterns also help keep everything in its proper place. However, they do require you to think a little more about how to achieve the ratios you want, and assembly can be a bit more complex than with simple gears.

Left
The planet gear sits between the sun and the ring and is held in place by them

Below
You can build planetary gears directly into wheel hubs



TUTORIAL



Macramé: A beginner's guide

Get creatively knotted with a very traditional craft that's well and truly back in vogue



Nicola King

@holtonhandmade

Nicola King is a freelance writer and sub-editor. Her favourite craft changes week by week, with her sewing machine currently in quilting overdrive.

Macramé was a very popular pastime back in the 1970s, which was perhaps a golden age of handicrafts of all kinds, when this form of creative knotting was used to fabricate lampshades, plant hangers, wall hangings, and many a bohemian beaded waistcoat with intricate and delicate patterning. This author fondly remembers her mum's extensive collection of 'Golden Hands' periodicals, an informative series of magazines that sought to share crafting goodness with the multitude. Macramé, in some form, always seemed to be in every single issue.

However, like many fibre-related crafts, macramé's history predates that particular era by some time, in fact by thousands of years. For example, some decorative carvings of the Babylonians and Assyrians



Above
There are many books available on the subject, and a pattern book of some kind is always useful as an inspiration source



YOU'LL NEED

- **3mm macramé cord** (e.g. **hsmag.cc/MacrameKit**)
- **Sharp scissors**
- **A tape measure**
- **A keyring** (with a D ring-shaped base – **hsmag.cc/DRingKRing**)
- **A comb for fringing** (optional)

Left
A few of the additional bits and bobs you can use within your work. The wooden beads and rings and dowels really complement the natural fibres being used

> The beauty of the craft is that there are no **hooks or needles required**

feature macramé-style knots. More recently than that, 19th century British and American sailors also made hammocks and belts from macramé, although back then it was called 'square knotting'.

Well, you can't keep a good craft down and, after falling out of fashion in the 1980s following its previous resurgence, macramé is popular once again, with millennials picking up on its relaxing benefits. You can find macramé kits in abundance in craft stores, and this author has even seen plant hanger kits for sale in supermarkets, and she's happy at the craft's revival. When she moved into her present family home many years ago, a macramé lampshade adorned the ceiling of one of the bathrooms – it went swiftly in the bin but now, as a fibre and textile crafts fan, she slightly regrets casting aside someone's vintage hard work.

Macramé is another calming and mindful hobby, and you can use it to create a huge variety of useful household items, as well as beautiful pieces of art and jewellery.

The beauty of the craft is that there are no hooks or needles required, as we are not weaving or knitting here – we are just knotting with our hands. Also, the emphasis is very much on using natural fibres such as cotton, jute, or hemp, for example – ideally these fibres are sustainable and plastic-free, so it can be argued that macramé is quite a 'clean' →



Left
A selection of macramé cords – obviously, as with any fibre craft, the thicker the yarn, cord, or rope that you use, the larger the finished product will be

TUTORIAL

QUICK TIP

Work at a clear, roomy table with good lighting so that you can clearly see what you are doing.



Figure 1
Attaching cord with the useful lark's head knot. This knot is also often used in jewellery making, and is also called a cow-hitch knot

WHAT CAN I MACRAMÉ WITH?

There are several options in terms of the fibres that can be used for this craft. Below are the main natural ones to consider, although you can also macramé with synthetic fibres should you so wish. We've opted to concentrate on natural fibres as they look great, are perfect for indoor projects, and they are biodegradable. Be aware that, if you are making projects that will sit outdoors with natural fibres, that biodegradability feature will mean that your makes will eventually degrade over time. For example, cotton will yellow in the sun, and jute and hemp will be affected by rain and sun – just something to bear in mind.

Clearly, the thickness of your material is a key choice, depending on what you are making. You'll need to consider the ply, i.e. how many strands have been twisted together to make a piece of cord or rope, and what effect does that give? Thicker fibres can be more challenging to work with, but certainly make a statement piece.

- **Cotton** – the world's most used natural fibre, it is soft and fairly strong too, and popular in the macramé world. It also gives a smooth finish and is the most versatile macramé fibre. Flexible and widely available, cotton is easy on the hands and won't stretch over time. It also comes in various thicknesses, from string to cord to rope, from 2/3 mm up to 40 mm.
- **Jute and hemp** – these are both fibres with low elasticity and a similar tensile strength. Jute is extracted from the stem of the jute plant, while hemp comes from the stalk of the hemp plant. Both of these can be more difficult to work than cotton and can be very rough on the hands, but they arguably have a very rustic and natural look to them, so it really depends on the aesthetic you are aiming for.
- **Linen** (in cord or yarn form) – perhaps not as common as other macramé materials, this fibre comes from the flax plant. More difficult to come across, but very strong, and often used for jewellery making.
- **Silk and leather cord** – both extremely strong macramé fibres, but perhaps less common. Leather cord is often used in jewellery making.
- **Natural yarns** – including merino wool, alpaca wool etc. Macramé knots will be much smaller if you use a natural yarn, however, as the fibre compresses so much.

hobby to pursue. Plus, if you decide to give it a try, you need very little in terms of equipment, so your craft space won't get unnecessarily cluttered and, perhaps most importantly, macramé is not hard to learn – once you have mastered just a handful of knots, you can make a plethora of items. So, in this tutorial, we are going to give you a basic macramé taster session.

LET'S GET TIED UP IN KNOTS

We're going to make a very simple keyring, and will be looking at just three basic knots: the lark's head knot, which simply attaches our cord to the keyring; the square-knot, which will create the main body of the piece; and the gathering knot, which will finish off our work.

STEP 1 CUTTING AND ATTACHING OUR CORD

A top tip is never to underestimate how much cord you are going to need. In fact, estimating how long to cut your cord length is arguably one of the most difficult stages. There's a tendency to cut cords too short, and you don't want to be joining cords halfway through your make. We'll cut two lengths of two metres each of our cord.

We then use the simple lark's head knot to attach the cord to the keyring. Take one length of cord – we've used 3 mm cord – and fold it in half. Take



Figure 2
A square-knot up close. We had to practise this a few times, but once we'd learnt it, we were away!

the looped end and thread it through the back of the D ring part of the keyring. Then, pull the cord ends through the middle of that loop in the cord (**Figure 1**). Repeat with the other length of cord. You should now have four lengths dangling from your keyring. We will be working with the two outer cords throughout. The two central cords are called the knot-bearing cords, and you should try and keep these two cords as straight and taut as possible.

> **There's a tendency to cut cords too short, and you** don't want to be joining cords halfway through your make

STEP 2 THE SQUARE-KNOT

Before we begin the main body, a word on tension. Everyone's tension is different, and we are aiming for a uniform, natural tension throughout. Don't pull the cords too tightly, but don't leave them too loose or you'll have huge gaps in your work – just pull them using a tension that feels natural for you and gives you the effect you want.

Take the left-hand knotting cord and take it over the two knot-bearing cords, and under the right knotting cord. Bring the right knotting cord to the left, under the two knot-bearing cords and over the left knotting cord (through the loop you've created). Pull the knotting cords, ensuring the knot-bearing cords are straight and taut. Now for the other side – bring the right knotting cord to the left, over the knot-bearing cords and under the left knotting cord. Finally, bring the left knotting cord to the right under the knot-bearing cords, and over the right knotting cord (through the loop). Pull to secure your square-knot (**Figure 2**). Repeat this until you have achieved the length you require.

RESOURCES

If you're interested in giving this craft a go, it's useful to do a little research before you start, and there are many books/websites that can answer any questions you may have and give you a dash of inspiration. Many modern macramé patterns have a contemporary twist so, don't worry, no 1970s throwbacks, unless of course that's the look you are aiming for:

- ***Macramé Pattern Book*** – Marchen Art – a handy guide that contains over 70 patterns that you can turn to for a multitude of projects
- ***Modern Macramé*** – Emily Katz – contains a whole heap of projects to get stuck into
- ***Macramé: Techniques and projects for the complete beginner*** – Sian Hamilton and Tansy Wilson – a great place to start for a newbie
- **YouTube – Majestic Macramé (youtube.com/@majestic.macrame)** – lots of very clear videos to help you on your macramé way
- **YouTube – Macramé School (youtube.com/@macrameschool/videos)** – make a macramé flower, make a macramé fish… it's up to you
- **Website: macrameuk.com** – includes a 'Knot bible' page for easy reference

QUICK TIP

When using natural fibres, ensure you store them well before you use them, wrapping them in something so they don't get too dry.

STEP 3 FINISHING OFF

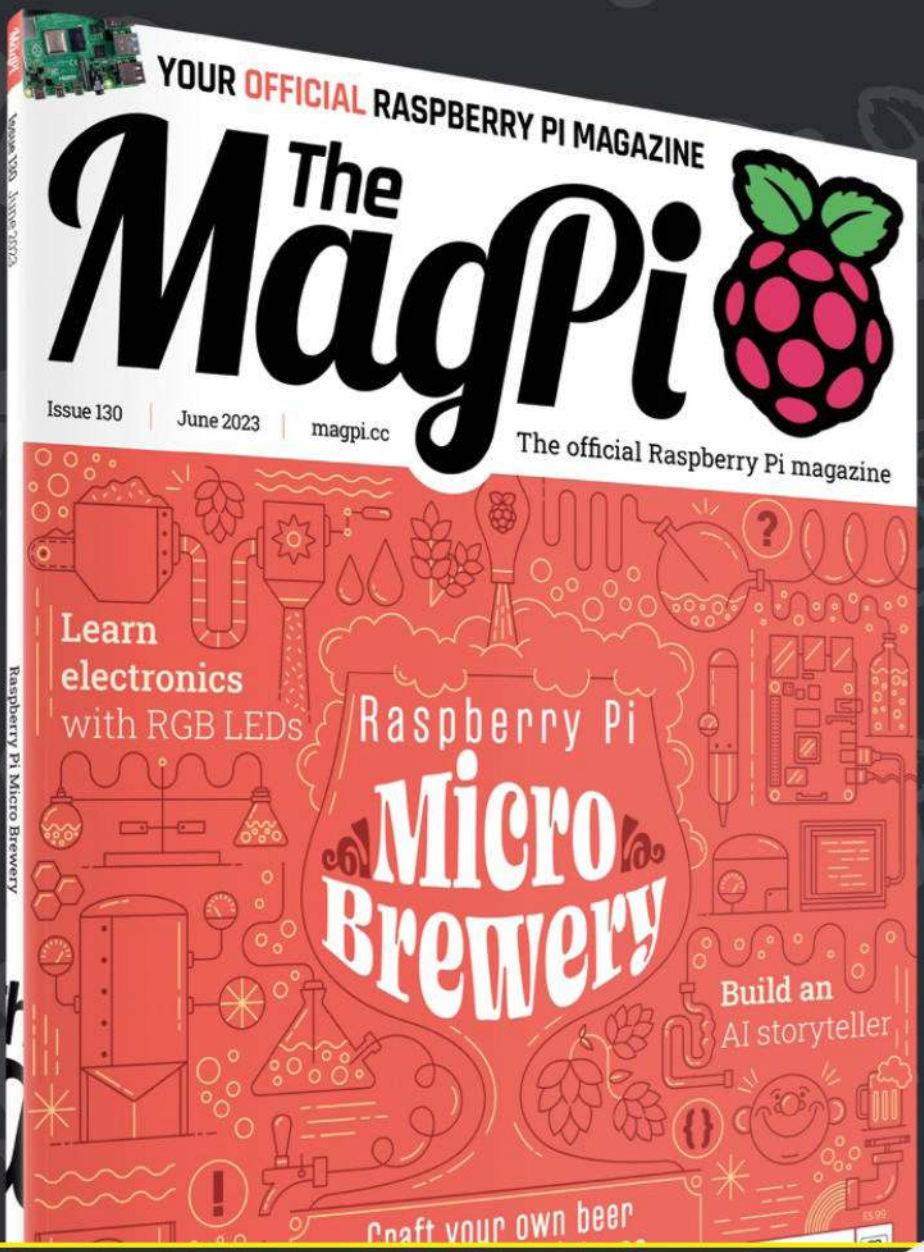
When you are happy with the length of your work, it's time to tidy up your ends. First, cut another length of cord, around 20 to 30 cm in length. Place the end of that against the loose cords, leaving a short end on the top, and form a loop, then wrap this new cord tightly around the bottom of your work several times. Then take the end of that cord and pull it through the loop you made (**Figure 3** uses a contrast colour to illustrate). Now, pull on the short end of the cord until the lower end of the cord is pulled up into the wrapped knot. Trim either end of that cord and tuck any loose ends into the wrapped knot. You can then either trim the remaining loose dangling ends to a length you like, or you can use a comb to turn them into a fringe, as we have done.

Once you've learnt a few knots, you can obviously gravitate towards more complicated macramé patterns, and maybe experiment with adding glass beads, shells etc. to create more elaborate and ornate pieces. We've also seen amazing work where artists have used a cotton hand dye and paintbrushes to colour their unique creations with a stunning ombré effect. So, get knotting!

Figure 3
A gathering knot to pull the ends of your work together



DON'T MISS THE **BRAND NEW** ISSUE!



SUBSCRIBE FOR JUST £10!

- **THREE!** issues of The MagPi
- **FREE!** Raspberry Pi Pico W
- **FREE!** delivery to your door



+ FREE RASPBERRY PI PICO W*

Three issues and free Pico W for £10 is a UK-only offer. Free Pico W is included with a 12-month subscription in USA, Europe and Rest of World. Not included with renewals. Offer subject to change or withdrawal at any time.

magpi.cc/subscribe

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated



PG 92

HEXTRACTION

3D-print yourself a new game

PG 86



BEST OF BREED

The best new maker products available now



PG 96

WYLDCARD

An e-paper card game platform for developers



ONLY THE BEST

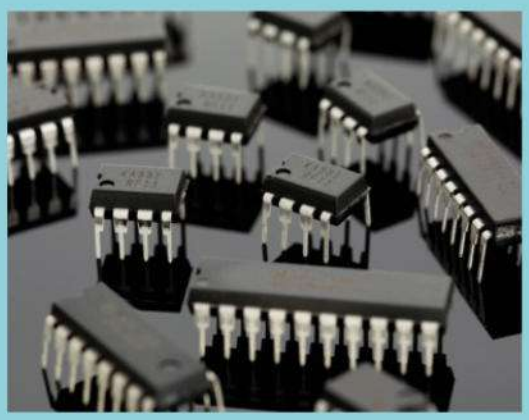
New and newish products that you can buy right now!

A collection of electronics for your next project

By **Marc de Vinck** @devinck

I've been writing these Best of Breed reviews for a while; at last count, it was over 60 issues covering over 350 products! With so many reviews, it can sometimes be challenging to come up with a new theme. At first it was easy to pick products, from displays and microcontroller boards to barometers and the hundreds of different sensors available for a Raspberry Pi. But when the supply chain issue hit the global economy, and so many developers went into hibernation, new products became much more difficult to find. Good news! Things have finally changed!

Only until recently have manufacturers, both large and small, been able to source components like the amazingly versatile RP2040 to create new boards and not worry too much about the supply. And common components required for products, like resistors and potentiometers, are now more readily available too. It's still not an ideal world when it comes to sourcing parts, and sadly it's going to be a while longer until you can just grab any microcontroller or single-board computer from a local store or website, but it's getting better. Everything is getting better. Finally!



So, in this Best of Breed, I'll be looking at some new and newish items hitting the market. There are a lot of great products out there, and they are in stock, ready to be soldered, tinkered, and integrated into your new project. Let's look at some of my favourite new – at least new to me – products.

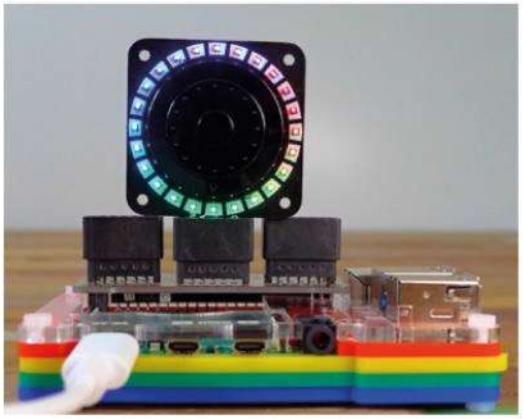
RGB Encoder Wheel Breakout vs Nicla Voice

PIMORONI $20.31 | pimoroni.com

ARDUINO $82 | store-usa.arduino.cc

Encoder wheels are cool, but RGB encoder wheels are really cool! Pimoroni has done a great job integrating a Nuvoton MS51 microcontroller and rotary scroll wheel to allow a lot of functionality to your project. This encoder features 24 RGB LEDs, directional four-way navigation, centre click and scroll wheel, all with nice clickety clicks. The encoder is easy to integrate into your next Raspberry Pi project thanks to the convenient I2C breakout.

The board is fully assembled and works with either 3V or 5V boards. It also includes a nice Python library and schematic, allowing for quick integration.



Left
Spinning wheels and flashing lights!



Above
Speech recognition, even when you're short on space

The small form factor Nicla Voice for Arduino allows anyone to easily implement always-on speech recognition into their next build, even a wearable. It features Syntiant's NDP120 Neural Decision processor running multiple AI algorithms and advanced machine learning. It's an amazing little device! And even more amazing is that most people, even beginners just starting out with an Arduino, could easily add voice recognition to their build. OK, maybe advanced beginners.

The Nicla Voice obviously includes a microphone, but it also has an integrated six-axis motion sensor and magnetometer, and on-board Bluetooth Low Energy (BLE) connectivity. It's a powerful little board that also runs in an ultra-low power mode, making 24/7 always-on sensor data processing possible.

VERDICT

RGB Encoder Wheel Breakout

RGB Encoder! What's not to love?

10/10

Nicla Voice

Amazing, but at a price.

9/10

New and newish products that you can buy right now!

BEST OF BREED



Inventor HAT Mini

PIMORONI $25 | pimoroni.com

I **really love all of Pimoroni's products. Each one is well thought out, always has great documentation, and includes easily understandable code.** The new Inventor HAT Mini is no exception. It's a versatile board that allows for simple motor, servo, and audio control for your Raspberry Pi. It's expandable, too, with its well-thought-out pass-through headers allowing you to stack on additional boards. It also includes four extra ADC-capable GPIO pins for adding more sensors. And if that's not enough, it also features eight addressable RGB LEDs for some variable feedback opportunities.

VERDICT

Inventor HAT Mini

A handy little board, making connecting servos easy!

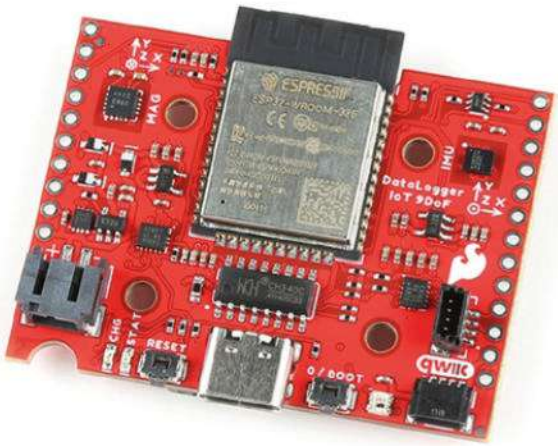
9/10

SparkFun DataLogger IoT – 9DoF

SPARKFUN $74.95 | sparkfun.com

SparkFun has another handy multipurpose breakout board: the DataLogger IoT – 9DoF. It's a utilitarian board with a utilitarian name. The board comes pre-programmed and will start to automatically output all the IMU, GPS, pressure, humidity, and distance sensors once powered up. You won't need to write any code, which can be good or bad depending on your background.

I think there are lots of people out there who just need the data and don't want to fiddle around in an IDE. If that's you, this is a great board. It was designed for users who just need to capture a lot of data to a CSV or JSON file and get back to their project. You can save the data to a microSD card or wirelessly to an IoT web service. A great solution for many people.



VERDICT

SparkFun DataLogger IoT – 9DoF

Useful and pre-programmed.

8/10

LOAD CELL

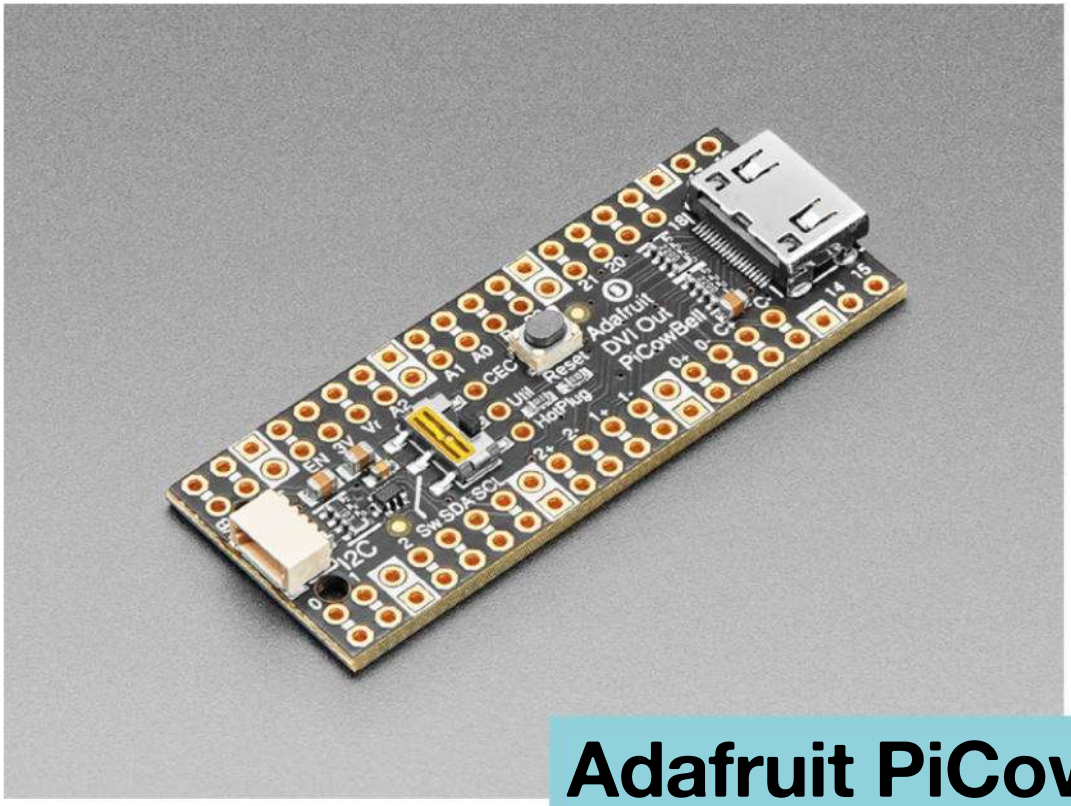
SPARKFUN $12.95 | sparkfun.com

Load cells are common when it comes to DIY electronic projects. Being able to precisely measure the weight of objects is easy, and the resulting data is much more precise compared to using an inexpensive piezo. Load sensors like this one – also called a strain gauge – can measure much heavier loads. I've come across many different types, but this one is unique, since it includes a means to hang the sensor and the object to be weighed. It's not overly complicated to build yourself, but it is nice to have it ready to go with the hooks and required extra hardware.



New and newish products that you can buy right now!

BEST OF BREED



Adafruit PiCowbell DVI Output for Pico

ADAFRUIT | $6.96 | adafruit.com

Adafruit has another uniquely named, and uniquely functioning, breakout board for the Raspberry Pi Pico – it's the PiCowbell. This little board allows you to display images and graphics from a Raspberry Pi Pico directly to an HDMI monitor or television. The board also features a Qwiic/STEMMA QT connector for fast I2C, a reset button and switch for rebooting or changing configurations. And did you see the price? Yeah, it's inexpensive, to say the least! I have a lot of projects that could use this handy little breakout board to make my life so much easier. If you need video, check out this board. Just note, it's video only, no audio. But it's still awesome!

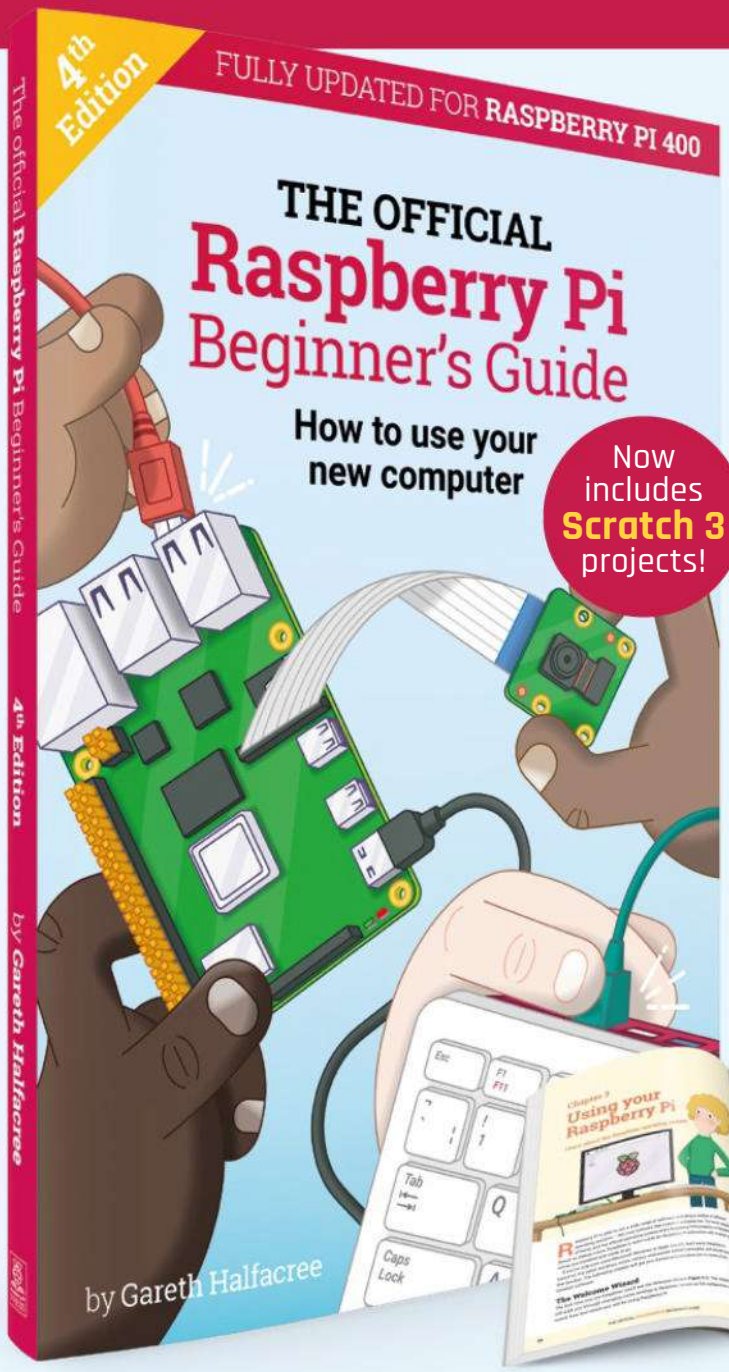
VERDICT

Adafruit PiCowbell DVI Output for Pico

If you need video from a Pico, you need this!

10/10

THE OFFICIAL Raspberry Pi Beginner's Guide



The only guide you need to get started with Raspberry Pi

Inside:

- Learn how to set up your Raspberry Pi, install an operating system, and start using it
- Follow step-by-step guides to code your own animations and games, using both the Scratch 3 and Python languages
- Create amazing projects by connecting electronic components to Raspberry Pi's GPIO pins

Plus much, much more!

£10 with FREE worldwide delivery



Buy online: magpi.cc/BGbook

Hextraction

Print your own puzzle game

playhextraction.com ◆ FREE | voidstar

By Ben Everard @ben_everard

Lots of makers create bits and pieces for games. We've seen miniatures 3D-printed, tables laser-cut, and dice created using almost every method conceivable. Hextraction is a little different because you can make the entire game.

You can 3D-print almost everything – the only things you have to buy are spools of filament and a handful of 10mm ball bearings. Some tiles might also require a bit of hardware, such as a bolt or magnet. It takes quite a bit of printing to get started (expect to keep your fabricator humming for a couple of days). It's a great way of using up odds and ends of filament because it looks great in a riot of colour.



Below ◆
Tiles can include multiple parts, like this two-piece teleporter tile

The game board is a slanted grid of hexagonal holes with on-ramps at the start and off-ramps at the end. Each player takes turns to put a tile in the board and roll a ball down an on-ramp (not necessarily in that order). If the ball gets stuck or falls off the board, then nothing happens (any stuck balls are left on the board whether they're stuck intentionally by a tile or by accident of a tile misfunction). When the board is full, then you can start destroying tiles, and if you run out of tiles so you can't play any more, the game's a draw.

So far, so straightforward. A lot of the tiles are simple directional tiles that take input from one direction and send it out another way. Some tiles use basic physics to introduce effects, such as the 'flip-flop' tile, which sends the ball out an alternate way each time, or the Pancho tile, which has a simple randomiser. Things get most interesting, though, with the special effects tiles – these have accompanying rules. For example, the teleporter set has 'in' and 'out' tiles, and when a ball goes into the in tile, you then start the ball again in the out tile.

SOLOTAIRE

There are also a couple of ways of playing the game by yourself – one where you take on a hoard of invisible monkeys (yes, really), and one where you create a random puzzle that you have to solve. Both of these use the same tiles and board as the multiplayer version, so you can print up a single set before deciding what to do.

Perhaps the most interesting part of this game – at least from a maker perspective – is that it's designed to be open source.

There's no fixed set of tiles that you have to use (though there is a recommended set for starting with), and you can simply print whichever ones you want to use. You can even take it further and design



Above
Simple physics effects – like this sozu tile, which only turns when the second ball hits it – can create fun gameplay

your own tiles for your games. There's a blank tile design available and you can modify this to create whatever tile you like, and share them with other players. Designing a tile isn't a completely trivial CAD task, but it's a decent starting point if you want to learn more and enhance your skills. It's something we'll be investigating in the next few weeks, and if there's interest, we'll do a tutorial on creating your own tile.

At the time of writing, the game is only a few weeks old, but there are already several third-party tile designs available. The Hextraction community is centred around the Thangs 3D design sharing website (**thangs.com**), and that's where you'll find the majority of the tiles. Obviously this does give players enough scope to create tiles that are terrible as well as great. The solution for this is that the game is democratic. At any point in the game, players can vote to get rid of a tile if they think it's having a negative effect on the game.



Above
The entire game can be 3D-printed, including the board and stand



Left
There are already lots of different tile designs to get you started

There are enough designs now to make a fun game, but it'll be interesting to see where the community takes this game. What new tiles can create new styles of gameplay? Time will tell.

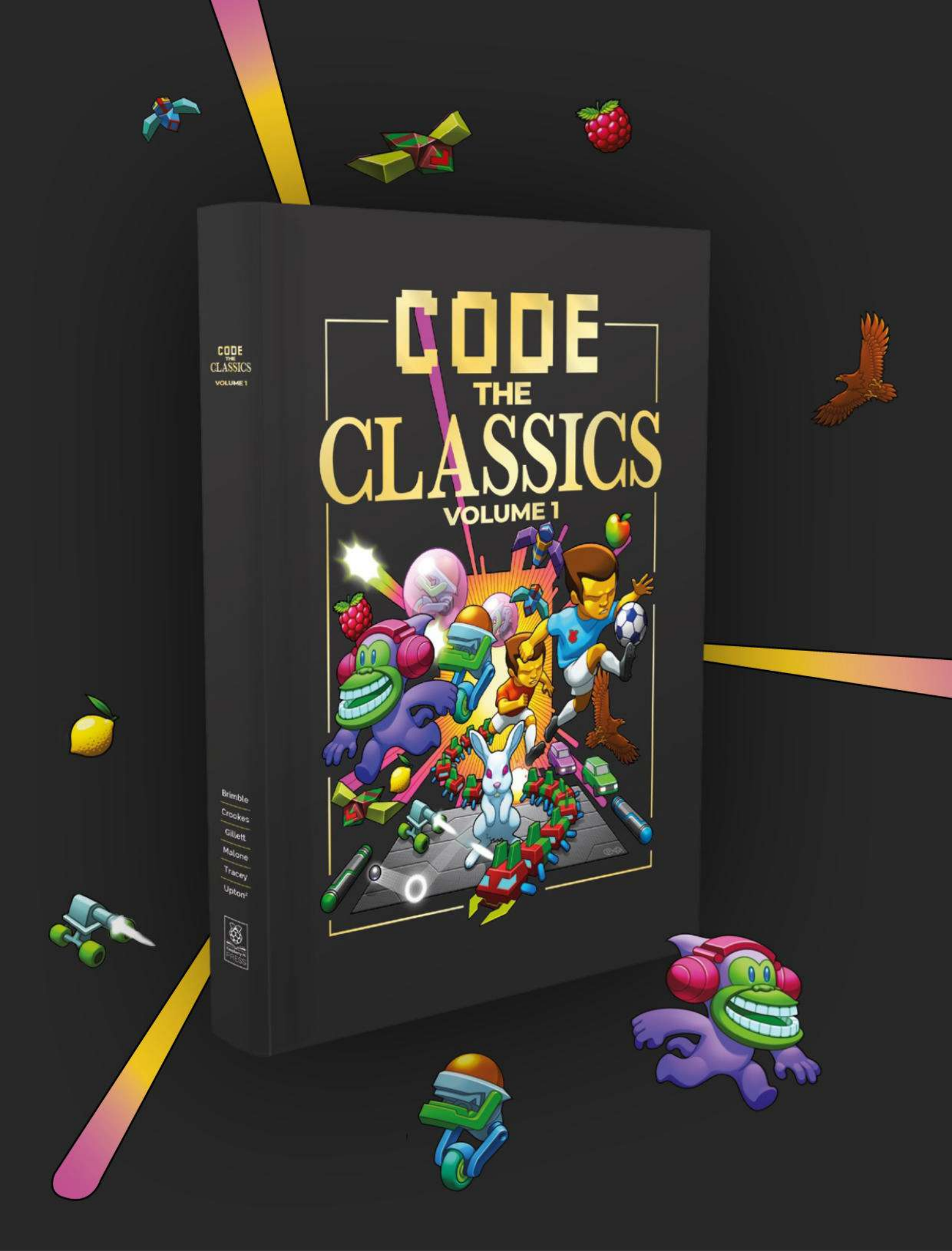
We've had a bit of trouble with a couple of tiles not performing quite as expected, particularly the 'Da Bomb' and 'Triple Trap'. It's hard to say if there's a fundamental issue with these tiles or if they just don't like the particular print settings we've used. We'll experiment a little, but no tile is essential to play, so if you do have a tile that's not performing as you'd like, you can just not include it in your game and print out some others. Other than this, it's been easy to play and print.

We've had great fun building and playing Hextraction. The game is a bit like a puzzle solver, and a bit like a collectable card game (only with tiles instead of cards). Exactly how the gameplay goes depends a lot on what tiles you have. Some people will prefer the simplicity of more physics-based games, while others will want to explore the full range of complexity that special effects can bring. It's entirely up to you (and the people you play with) what tiles you include. ◘

VERDICT

A fun 3D printing project with scope to help you learn 3D design.

9/10


CODE
THE
CLASSICS
VOLUME 1
Brimble
Crookes
Gillett
Malone
Tracey
Upton²





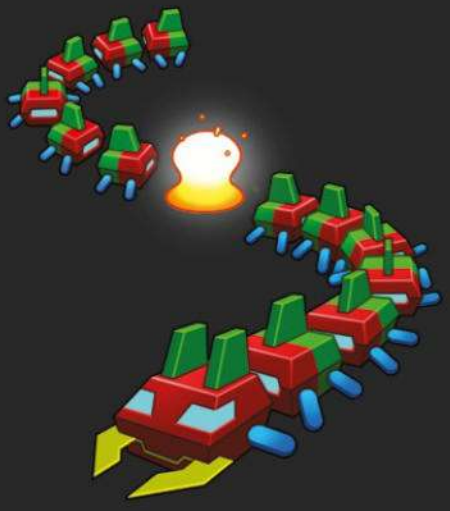
CODE THE CLASSICS VOLUME 1

This stunning 224-page hardback book not only tells the stories of some of the seminal video games of the 1970s and 1980s, but shows you how to create your own games inspired by them using Python and Pygame Zero, following examples programmed by Raspberry Pi founder Eben Upton.





- ***Get game design tips and tricks from the masters***
- *Explore the code listing and find out how they work*
- ***Download and play game examples by Eben Upton***
- *Learn how to code your own games with Pygame Zero*



Available now **hsmag.cc/store**

CROWDFUNDING NOW

Wyldcard DevKit

The card game that changes as you play it

From $270 | **hsmag.cc/wyldcard** | **Delivery:** Nov 2023

Wyldcard is a system for building card games, except, unlike regular card games, the cards can change as you play them. E-paper allows a console (known as a Plinth) to send new images to the cards so that they can adapt to the gameplay. This could be aesthetic (such as showing your characters sustaining damage), or alter the way the card is used in the game. As the name suggests, this is a DevKit for people looking to make games (the team have created a JavaScript API to help you work with the hardware). At present, there are no games available, so it is down to you to build something interesting.

While this does look cool, we find the cost proposition a bit strange. A set of three e-paper cards for $60 is on the steep side, but you're paying for developer time and polish, so it doesn't seem unreasonable. However, a DevKit (which is needed to use the cards) is $270 – we can knock off $60 for the price of the included cards, so that's $210 for a Raspberry Pi Zero, an enclosure, and a PCB to mount the cards on. That feels a bit pricey. Perhaps that could be justified if you could then play a game, but you need two of these for a two-player game, so you're over $500 (including the cards) in total before you even start playing. Again, this could perhaps be justified if it was paying for developer time to make the game, but this doesn't include any games – it's the developer kit for you to create your own games.

Despite all this, the hardware does look like it's well-thought-out, and there are probably some interesting game dynamics possible with this style of setup. Is it worth the money? That's up to you.

BUYER BEWARE

When backing a crowdfunding campaign, you are not purchasing a finished product, but supporting a project working on something new. There is a very real chance that the product will never ship and you'll lose your money. It's a great way to support projects you like and get some cheap hardware in the process, but if you use it purely as a chance to snag cheap stuff, you may find that you get burned.





next month

issue #69 ON SALE 27 JULY

SUMMER PROJECTS

ALSO

- RASPBERRY PI
- 3D PRINTING
- LASER CUTTING
- MUSIC
- AND MUCH MORE

DON'T MISS OUT

hsmag.cc/subscribe



Cha-ching

Franz's coin sorting machine, which he's kindly shared with the world on **Printables.com**, is a great example of the way that different materials can mess with our sense of scale. Wood, with its user-friendly grain feature, gives you an at-a-glance idea of how big something is. The tiny imperfections in concrete, the crystals in a block of stone, the fibres in wool, all serve to lend a familiarity to objects that we've not seen before.

This is different. We can honestly say that the first time we looked at this machine, we thought it was the size of a rollercoaster ride. 3D printing has been around for a while now, but it still counts as new in the human tradition of building things. The rule book hasn't been written yet, so go forth: play, fail, and learn.

HiPi.io

PiKVM

Manage your servers or workstations remotely

A **cost-effective** solution for data-centers, IT departments or remote machines!



PiKVM HAT
for DIY and custom projects



Pre-Assembled version

- Real-time clock with rechargeable super capacitor
- OLED Display
- Bootable virtual CD-ROM & flash drive
- Serial console
- Open-source API & integration
- Open-source software

Available at the main Raspberry Pi resellers

core ELECTRONICS · PiShop.ca · RPishop.cz · Welectron. · elektorstore · PIMORONI

Pi-Shop.ch · ThePiHut · PiShop.us · CloudFree · PiShop.mx

Reseller suggestions and inquiries:
wholesale@hipi.io